PracStat: A voice-activated statistic tracking software for the Ball State men's volleyball team

An Honors Thesis (HONR 499)

by

Adam Wessel

Thesis Advisor

Dr. Dolores Zage

Ball State University

Muncie, Indiana

May 2019

Expected Date of Graduation

May 2019

Abstract

In the world of sports, keeping accurate and meaningful statistics is crucial for success. Coaches rely on this information to make important decisions. As a former member of the Ball State University varsity men's volleyball team, I noticed that the statistic tracking methods used by our coaches was outdated and error-prone. Therefore, I was motivated to create a new and improved software that would greatly improve upon the prior methods. This software project was the creation of PracStat – a voice activated stat-tracking software system designed specifically for the Ball State men's volleyball program. As the project manager in charge of PracStat, it was my job to drive the project from inception to completion and to lead the team throughout the entire software development life cycle.

Acknowledgements

I would like to thank Dr. Dolores Zage for advising my team and me throughout the duration of this project. The help and guidance she provided was invaluable and significantly impacted the final outcome of PracStat.

I would also like to thank Joel Walton, the Ball State men's volleyball head coach, who acted as our client for this project. He motivated us to create something special that will truly make an impact on the volleyball program.

I would like to thank Dexter Edwards and Alex Bobay, my team members, both of whom spent countless hours helping to make PracStat the best it could possibly be.

Table of Contents

Process Analysis Statement

When thinking about a project to carry out for my honors thesis, I knew I wanted it to relate closely to my major – computer science – as well as my future work as a project manager of software development projects. This way, I would gain valuable experience in all phases of the software development lifecycle and enter the workforce with a newfound confidence and knowledge. However, I also knew that all software projects require a team effort to be successful. Therefore, I decided it would be best to combine my honors thesis with my computer science capstone (CS 495 and CS 498) where I met my team members Dexter Edwards and Alex Bobay.

Our first task was to find a client who could benefit from a software project. As a former member of the Ball State University varsity men's volleyball team, I noticed right away that our coaching staff used very outdated statistic-tracking methods that were time-consuming and inaccurate. Dexter, Alex, and I (who called our team W.E.B. Enterprises) then met with head coach Joel Walton to talk about his need for a new and improved stat-tracking software. Joel was thrilled about the possibility of this software, and quickly agreed to act as our client throughout the entirety of the project.

As a team, we talked to Coach Walton about his current methods of keeping statistics and how we could streamline this process for the team managers. Before PracStat was developed, the Ball State men's volleyball team managers took statistics manually by keeping tallies on a clipboard. This required at least three people to keep statistics that were still not as accurate as they could be. Coach Walton introduced the idea of voice recognition as a method of tracking statistics which my team found very interesting. We decided to pursue that idea.

After getting a general idea of what the project would consist of, it was time to ensure that the project was feasible. Dexter, Alex, nor I had ever developed a software as complex as voice recognition. We had to make sure that PracStat would be a project we could develop on time and under budget. We researched countless other statistic-tracking systems to get an idea of what PracStat could look like. We used widely accepted feasibility estimations including the Delphi Estimation and Function Point Estimation to get an idea of how in-depth the project could be. We decided that a stat-tracking software that utilizes voice recognition would be possible. We also decided to utilize a mixture of waterfall and agile software development methodologies to keep our project moving forward as smoothly as possible.

The next phase consisted of user story/use case development and our preliminary design. During this phase, we needed to think about who the users of PracStat would be and how we could design a system that would be intuitive for them to use. We developed a total of eleven use cases that would be referred back to throughout the entire development process. It was during this phase that my group started meeting much more regularly. I took on the project manager roll (in addition to many other roles), and so it was my job to organize group meetings, assign tasks, make sure milestones are met, and drive the project to completion. My group members Dexter and Alex were knowledgeable and very motivated to create a great project which made my job that much easier.

Once we had an overall idea of what our project might look like, it was time to specify the requirements of our project with our client. The result of this was a Software Requirements Specification document in which every requirement (both functional and nonfunctional) was specified and quantified clearly. Successful completion of these requirements would mean that PracStat is a successful project overall.

W.E.B. Enterprises then started writing the PracStat code. Each team member developed a significant amount of the overall code. My main assignment was to code the voice recognition aspect of PracStat. Since our team was most familiar with the Java programming language, I utilized the Sphinx4 Java library to incorporate the voice recognition functionality. Along with writing the code, we needed to thoroughly test it with a multitude of quality assurance and testing techniques ranging from unit testing, performance testing, stress testing, usability testing, and more.

Finally, we delivered the final PracStat product to our client – Joel Walton. Coach Walton was very impressed, especially with the voice recognition feature and the spreadsheet generation feature. He plans to try using it in preseason next Fall.

Development Manual
for

PracStat

Version 1.0 approved by Adam Wessel, Project Manager, W.E.B. Enterprises on May 1, 2019

Prepared by Adam Wessel, Alex Bobay, Dexter Edwards

W.E.B. Enterprises

May 1, 2019

Introduction

This Development Manual will outline the development process of PracStat from inception to completion and delivery. PracStat was designed and created by W.E.B. Enterprises consisting of team members: Adam Wessel, Alex Bobay, and Dexter Edwards. PracStat is a voice activated stat-tracking software application for volleyball, designed specifically for the Ball State University men's volleyball team coached by our client - Joel Walton. This manual is intended primarily for Dr. Dolores Zage at Ball State University, but also for anyone interested in how PracStat came about. The manual details every step of the design and development process for PracStat version 1.0. Detailed development information for PracStat versions later than PracStat 1.0 are currently out of the scope of this document.

Team Members
Adam Wessel

Alex Bobay

Dexter Edwards

Original Proposal

The Ball State men's volleyball team, coached by Joel Walton, is searching for an alternative stat recording system. None of the programs currently used satisfy all the needs expressed by the head coach. Some can't support more than 14 people, making it less useful for Ball State's 20-person team. Others involve complicated codes that require a trained statistician to use. There is also one called TapStats which looks nice but is not able to fit all of the stats in one page without scrolling to the side. An improved, easy to use program would help Mr. Walton out by allowing him more face to face time with the team during practices. It would also remove the extra time that Mr. Walton would normally have to spend transferring the collected stats to an Excel spreadsheet. By offering an improved system, Ball State could potentially save money by spending less on people to keep stats and by not having to purchase a wide range of programs and apps.

System Overview

PracStat is a self-contained application running independently on a team device. If prompted, the application generates a .csv file containing information tracked within the app. Using java file creation, we are able to customize and create unique stat spreadsheets catering to individual players and coaches.

Assumptions and Dependencies

The Ball State men's volleyball team will need to have access to a laptop in order to record stats during practices. According to Coach Walton, this will not be a problem. The laptop will also need to be microphone-compatible in order to process the voice commands.

All users of PracStat should have a basic understanding of how to use a computer. Users will also need to know how to configure and use an external microphone in order to record stats orally, as the microphone will need to be enabled to record voice statistics. PracStat is intended to run only on Windows.

Feasibility Study

Executive Summary

W.E.B. Enterprises is a startup software development team that exists to develop software primarily for use in the realm of sports. Currently, we are in the process of designing a new system of stat recording software for the Ball State men's volleyball team and head coach Joel Walton. Looking at the current market of stat recording applications, there is a significant lack of quality stat-tracking applications that have voice-recognition capabilities. We think that our approach of utilizing this feature in our software will set our product apart from most of the applications that are currently on the market. Additionally, we think that our product will be important in improving the BSU volleyball program in terms of analyzing stats from practices.

Overall Description of Products and Services

The Ball State men's volleyball team, coached by Joel Walton, is searching for an alternative stat recording system. None of the programs currently used satisfy all the needs expressed by the head coach. Some can't support more than 14 people, making it less useful for Ball State's 20-person team. Others involve complicated codes that require a trained stat-keeper to use. There is also one called TapStats which looks nice but is not able to fit all of the stats in one page without scrolling to the side. An improved, easy to use program would help Mr. Walton out by allowing him more face to face time with the team during practices. It would also remove the extra time that Mr. Walton would normally have to spend transferring the collected stats to an Excel spreadsheet. By offering an improved system, Ball State could potentially save money by spending less on people to keep stats and by not having to purchase a wide range of programs and apps.

At the moment, there are two main scenarios that we are envisioning for how we will implement this project. We are focusing now on developing a desktop application which will be able to be used with Mr. Walton's laptop. Our other scenario is developing a mobile application for iOS devices that Mr. Walton would use on his iPad and iPhone. Out of the three of us, no one has any experience developing mobile applications. Because of this, we feel that it would be much more feasible to implement this project as a desktop application, which we are all very familiar with from past classes. In a desktop application, it will be much easier to generate an Excel spreadsheet from the stats data. There is also an open-source Java library called sphinx4, which will be key in getting the voice recognition aspect of the project working. We think that integrating this open-source library will be better than coming up with our own implementation of voice recognition. This will give us more time to focus on what we do with the stats that are entered instead of worrying about how to get them in the first place. Once we have the stats stored in the program, we plan to generate an Excel spreadsheet which will show the stats in an intuitive way.

Assuming that we are able to successfully implement a desktop application with all of these features, we plan to see if it would be feasible to convert it to a mobile application with any remaining time that we have left. At this point we have not really done much research about converting desktop applications to a mobile format. We would save this endeavor for a potential

second or third iteration, assuming that there is enough time. For now though, we will keep our focus on developing a desktop application.

Technology Considerations

The Ball State men's volleyball team will need to have access to a laptop in order to record stats during practices. According to Coach Walton, this will not be a problem. The laptop will also need to be microphone-compatible in order to process the voice commands.

Additionally, there will not be any added costs with switching to a new stat-recording system. At this time, Coach Walton is prepared to switch to a new system and is not tied down to any one application. This will ensure that there is a smooth transition between systems.

Client Partner Communications

Throughout the duration of this project, all communication regarding status, updates, deliverables, etc. will be directed toward Joel Walton whose contact information is listed in the communications sheet linked to below. Coach Walton will also be the client partner with the authority to sign off on any and all project milestones. Communication with the client partner will take place both in-person and over email. All formal presentations with the client partner will take place at in-person meetings that will be scheduled prior to the event. All in-person communication will be recorded in a table at this link. For more basic status updates, email will be our primary method of communication. Email and/or phone will also be used to schedule the in-person presentations. We have also provided Joel Walton with the contact information of each of our team members so that he can contact us with any questions or concerns that might come up.

Communication between our group members will also take place both in-person and over email/phone. We have weekly meetings scheduled for every Sunday around 8:00 PM during which we can work on the project together. We are also utilizing a mobile application called GroupMe to conveniently send/receive text messages between group members. Additionally, we will be using both Google Drive and Agantty to manage our project and stay organized.

Product/Service Marketplace

The marketplace for volleyball stat tracking applications is not particularly large, and for this reason, we feel that we will have a unique opportunity to offer new features that are not currently being used by any other potential competitors.

At the moment, there are several free and paid volleyball stat-tracking apps in both the Apple App Store as well as the Google Play App Store. However, we are focusing on iOS apps so we will not be concerned with Android apps. Overall, these apps are generally simple in nature and have basic stat tracking and visualization tools. A few of them also offer advanced data analysis and do more than the standard free app. Some of the top apps include iStatVball 2, Volleyball Ace Stats, Volleyball Stat!, and SoloStats Touch Volleyball.

We will be primarily interested in developing a desktop application which could potentially be in competition with some of the mobile applications listed above. Our goal though is to focus mainly on recording stats and generating a spreadsheet with the results. We do not

plan on offering any other major features that one might see with larger services such as team organization. One good service that offers stats recording as well as other team management services is TeamSnap. Overall, this service does more than what we are planning to accomplish with our application. However, we are going to be focusing on recording stats via voice input, which is not a feature that TeamSnap offers. We feel that trying to develop an alternative to TeamSnap with all of its features would not be practical in the time that we have to work on this project.

Outside of iOS apps, the volleyball team uses VolleyMetrics to analyze game stats. This service is ideal for games but not practices, which is where our product will fit in. As a whole, our product will be an additional tool to be used alongside VolleyMetrics. For this reason, we do not consider VolleyMetrics to be a competitor.

We feel that users of our software will be drawn to our unique features including voice recognition as well as the ability to analyze more people per team than our competitors offer in their apps. These two limitations are some of main reasons why Coach Walton is looking for an alternative system to keep track of practice stats. At the moment, Coach Walton is currently using two iOS apps: Tapstats and iStatVball2. Despite using these applications, Coach Walton still has to spend extra time converting the data stored in these apps to an Excel spreadsheet format in order to analyze them and post them in the locker room. Our application would take care of this for him so that he is not wasting extra time that could be spent doing something more productive.

Marketing Strategy

At this time, we are not planning to market our project very heavily. Given the low budget of the project, the goal for marketing will be for the program to be good enough to be spread by word of mouth, with coach Walton recommending the program to other coaches he is in contact with. Further promotion could be done by contacting individual volleyball organizations by email. We think that our software will stand out from our competitors through our voice-recognition feature and the feature of exporting the data as an Excel spreadsheet.

Organization and Staffing

Switching to a new statistic tracker will not really affect the organization and staffing of the BSU volleyball team at all. The new tracker will be used exclusively for practices, so the advanced stat tracking from VolleyMetrics that doesn't get done until after games will still exist. At most, the new program will eliminate the need for one person from the previously two-person stat tracking team. No new people will need to be hired to support usage of the program, as the goal is for it to be intuitive enough to be used by one person with minimal explanation.

Effort
Delphi Estimation:

Name of Expert	Domain Expertise		Development Platform Expertise	
	Expertise Description	**Number of Years**	**Expertise Description**	**Number of Years**
Dexter Edwards	General stats	3	Java, C#, Python	4
Adam Wessel	Volleyball-specific expert	8	Java, Python	4
Alex Bobay	General stats	3	Java, Ruby, Python	4

Name of Expert	Size (Hours)
Dexter Edwards	300
Adam Wessel	415
Alex Bobay	345

The final convergence of the Delphi Estimation is 353 project hours.

Function Point Estimation:

# of user inputs	10	X	3	4	6	4	=	40
# of user outputs	10	X	4	5	7	5	=	50
# of user inquiries	5	X	3	4	6	4	=	20
# of files	0	X	7	10	15		=	0
# of external interfaces	1	X	5	7	10	7	=	7
						Count-total =		117

Rate each factor on a scale of 0 to 5:	0 - No Influence	1 - Incidental	2 - Moderate
	3 - Average	4 - Significant	5 - Essential

1. Does the system require reliable backup and recovery?	4
2. Are data communications required?	0
3. Are there distributed processing functions?	0
4. Is performance critical?	3
5. Will the system run in an existing, heavily utilized operational environment?	0
6. Does the system require on-line data entry?	4
7. Does the on-line data entry require the input transaction to be built over multiple screens or ope	0
8. Are the master files updated on-line?	0
9. Are the inputs, outputs, files, or inquiries complex?	2
10. Is the internal processing complex?	0
11. Is the code designed to be reusable?	4
12. Are conversion and installation included in the design?	0
13. Is the system designed for multiple installations in different organizations?	5
14. Is the application designed to facilitate change and ease of use by the user?	5
sum of Fi =	27

Funtion Point Metric = count-total * [.65+.01*sum Fi]

= 108

Software Development (Elaboration and Construction)

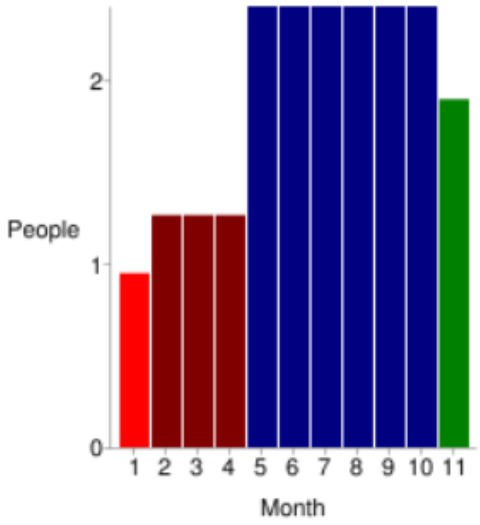
Effort = 19.2 Person-months
Schedule = 9.7 Months
Cost = $0

Total Equivalent Size = 5724 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.2	1.2	0.9	$0
Elaboration	4.6	3.7	1.3	$0
Construction	14.6	6.1	2.4	$0
Transition	2.3	1.2	1.9	$0

Staffing Profile



Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.2	0.6	1.5	0.3
Environment/CM	0.1	0.4	0.7	0.1
Requirements	0.4	0.8	1.2	0.1
Design	0.2	1.7	2.3	0.1
Implementation	0.1	0.6	5.0	0.4
Assessment	0.1	0.5	3.5	0.6
Deployment	0.0	0.1	0.4	0.7

As the above effort estimations demonstrate, development of the PracStat application is a small to medium-sized effort. PracStat is a project that will take the entire two semesters to complete, but it is also small enough in scope that we should (based on our estimates) be able to successfully complete it on time and under budget.

Schedule

This project is expected to take roughly 8 months to complete (August 2018 - April 2019). The following is a high level schedule of some significant milestones for this initiative during this semester:

August 21, 2018: Initiate project

September 13, 2018: Receive project approval

September 27, 2018: Complete DFD assignment

October 25, 2018: SRS first draft due

November 29, 2018: SRS review due

During the 2019 Spring Semester, we will begin the process of implementing and testing our project based off of planning we are doing right now. A more detailed schedule including all tasks and deliverables will be created during the planning phase of this project.

Financial Projections

Financial benefit is not a key goal of this project as we are not expecting to make any kind of financial gain. However, the creation of a complete and intuitive stat-tracking application has the potential to make money. If we are able to end up converting our application to mobile format, we could possibly release it on the Apple App Store for a potential profit. The main factor driving the successful completion of this project is our final grade in the CS495 capstone course. A good, passing grade in this course will directly affect each of our members' ability to make money in the future. It will also provide us with valuable software development experience. Additionally, this experience will be valuable in giving us an idea of what to expect when working with clients in a real-world setting as opposed to just doing a project for a professor.

Findings and Recommendations

Based on the information presented in this feasibility study, it is recommended that W.E.B. Enterprises approve PracStat and begin project initiation. The findings of this feasibility study show that this project will greatly benefit the Ball State men's volleyball program (as well as other volleyball teams) and has a good probability of success. The key findings are as follows:

Technology:

- Will utilize existing technology/software which lowers project risk
- This technology is simple to maintain and operate with a relatively easy learning curve compared to previously used applications

Organizational:

- We have a good relationship with our client due to previous interactions
- We have scheduled class time together and additional weekly meetings

Workload:

- Do not need to hire or recruit any additional human resources to work on this project
- Our estimations indicate that this project has a good chance of success given that the workload will be split among three people and we will each have our own responsibilities

Financial:

- We currently do not have plans to profit financially from this project specifically
- We are utilizing this project as a means to get some practical experience so that we can open up more opportunities for jobs in the future

Software Requirements Specification

Purpose and Scope

The Ball State men's volleyball team, coached by Joel Walton, is searching for an alternative stat recording system. None of the programs currently used satisfy all the needs expressed by the head coach. Some can't support more than 14 people, making it less useful for Ball State's 20-person team. Others involve complicated codes that require a trained statistician

to use. There is also one called TapStats which looks nice but is not able to fit all of the stats in one page without scrolling to the side. An improved, easy to use program would help Mr. Walton out by allowing him more face to face time with the team during practices. It would also remove the extra time that Mr. Walton would normally have to spend transferring the collected stats to an Excel spreadsheet. By offering an improved system, Ball State could potentially save money by spending less on people to keep stats and by not having to purchase a wide range of programs and apps.

In Scope:

- Phase 1
  - Voice input
  - Excel spreadsheet output
  - Team sizes of 20+
  - Basic stat keeping capability
- Phase 2
  - Different user privilege levels (coach and player)
  - Selecting specific stats to include (or exclude) on excel spreadsheet output

Out of Scope:

- Phase 3
  - Handle simultaneous practices
  - Email Excel spreadsheet

Product/Service Description

Collegiate volleyball coaches utilize statistics for scouting, tracking, and strategic purposes. Consistent, accurate statistics are necessary for a team to be successful. However, coaches cannot devote all of their time to recording team and individual statistics. Coaches also don't have a team of managers to track statistics. Therefore, a stat-recording system that is accurate and requires only one manager would be very beneficial. Since collegiate volleyball teams consist of more players than most other volleyball teams, the stat-recording system needs to be able to handle twenty or more players.

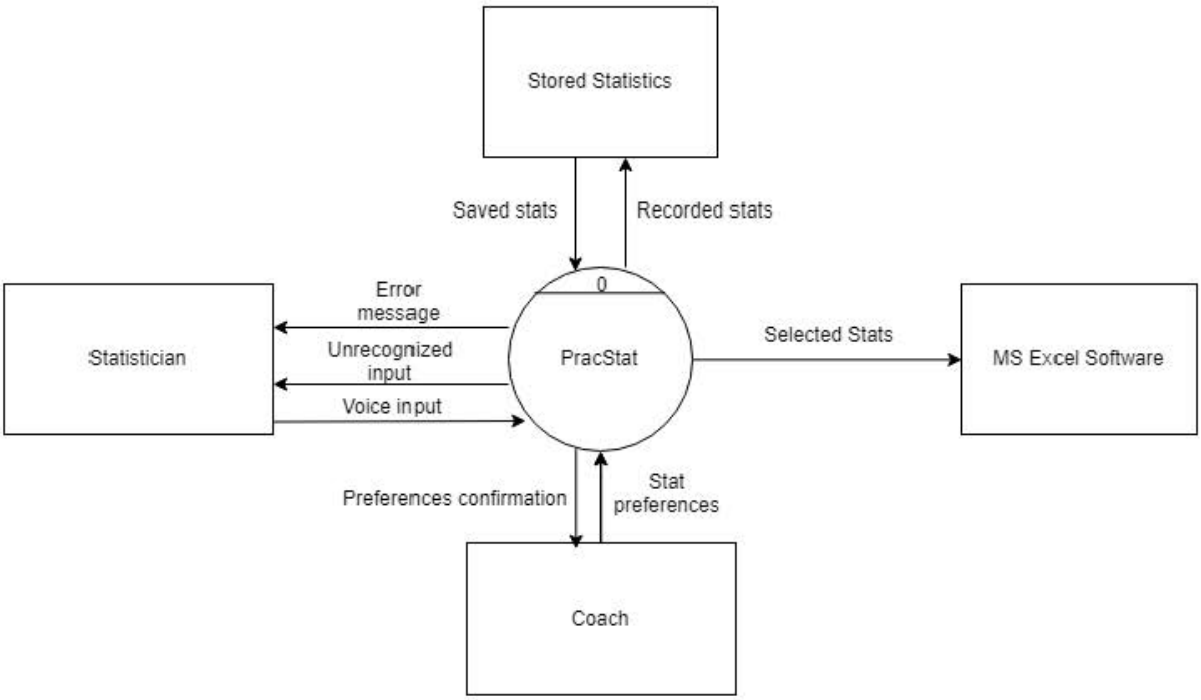
Product Context:

PracStat will be a self-contained application running independently on a team device. If prompted, the application will generate an excel .xlsx file containing information tracked within the app. Using the Apache POI libraries, which read and write files in Microsoft Office formats, we will be able to customize and create unique stat spreadsheets catering to individual players and coaches.

Context-Level Diagram:

Adam Wessel, Dexter Edwards, Alex Bobay

Context-Level Diagram



Data Dictionary

Statistician - any person that uses the program to record stats

Unrecognized input - input given by statistician that is not recognized by PracStat

Voice input - input given by statistician via microphone

Error message - message returned by PracStat when an error occurs

PracStat - the name of our program/process

Coach - the head coach of the team

Stat Preferences - specific stat categories chosen by the coach to appear in the Excel spreadsheet output

Preferences confirmation - message returned by PracStat to confirm the selected stat preferences

MS Excel Software - displays selected stats

Stored Statistics - file(s) that contain all recorded practice stats

Recorded Stats - the "raw" statistics that the statistician keeps during practices

Saved Stats - stats that are kept in the Stored Statistics file(s) to be used/viewed later

Selected Stats - a subset of statistics that are chosen to be displayed based on the stat preferences

User Characteristics:

The following are general customer profiles for each type of user who will be using the product.

Statistician:

- Knowledgeable about volleyball statistics and what plays add to specific stats (e.g. Digs are given only when players receive an attacked ball and it is kept in play, not when a ball is brought up off a blocked ball).
- Able to speak English loudly and clearly into a laptop microphone.
- Experienced with watching, stat-tracking, or playing volleyball.

Coach:

- Experienced in analyzing and strategizing in the game of volleyball.
- Has advanced knowledge about both team and individual statistics.
- Knowledgeable about calculated volleyball statistics (e.g. hitting percentage, assist percentage, etc.)

Player:

- Has a good deal of experience playing the sport of volleyball.

Assumptions:

The Ball State men's volleyball team will need to have access to a laptop in order to record stats during practices. According to Coach Walton, this will not be a problem. The laptop will also need to be microphone-compatible in order to process the voice commands.

All users of PracStat should have a basic understanding of how to use a computer. Users will also need to know how to configure and use an external microphone in order to record stats orally. PracStat is intended to run only on Windows.

Constraints:

- The system's output statistics spreadsheet needs to resemble the official NCAA box score spreadsheet.
- The system must run on a windows laptop with a standard microphone.
- User privileges (coach and player)
- All stats must be displayed on one page
- Configuration files will be generated and updated by the system
- The system will be written in Java

Dependencies:

- The laptop's microphone must be enabled for the system's voice recognition feature to work.

Requirements

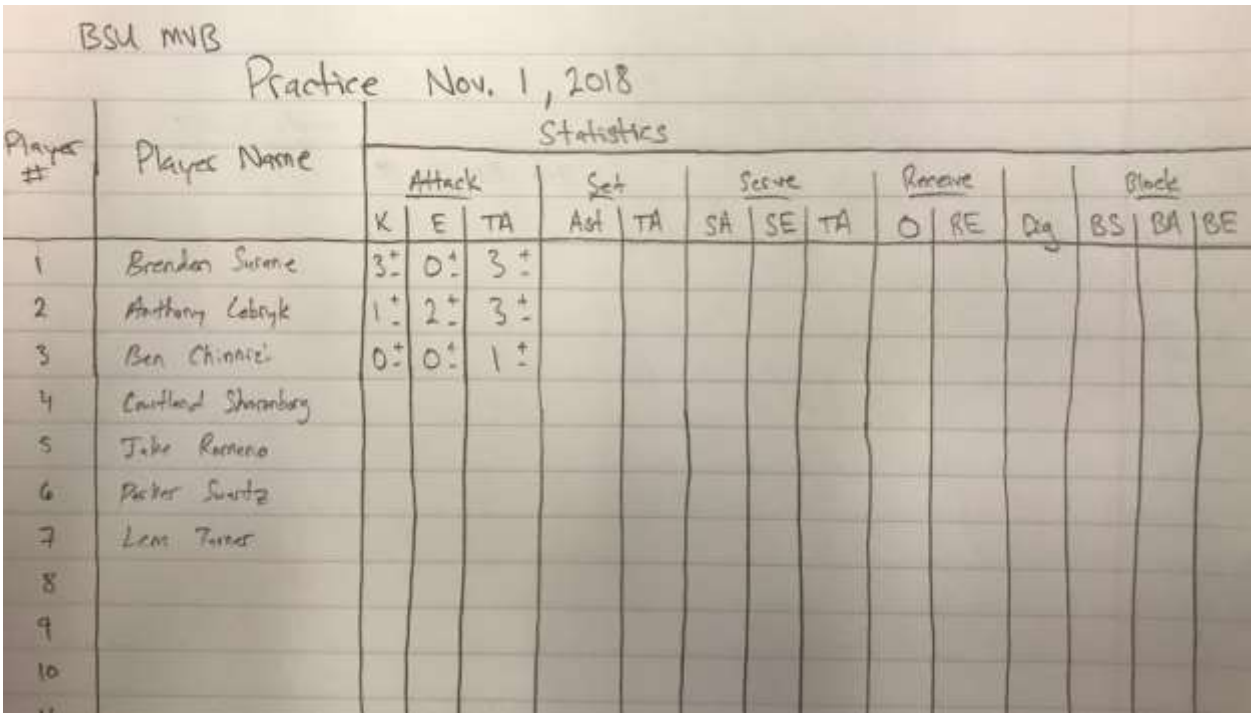
Functional Requirements:

Req#	Requirement	Comments	Priority (1 is Highest)	Date Rvwd	Client Reviewed / Approved
FR_LR_01	The system should allow access based on individual role on team (coach, statistician, or player).	Use Cases: 2, 3	2	10/3/18	Joel Walton
FR_LR_02	The system should store all recorded statistics in a file.	Use Cases: 1, 8, 10, 11	1	10/3/18	Joel Walton
FR_LR_03	The system should track the statistics of at least 20 players per team.	Use Cases: 2, 3	3	10/3/18	Joel Walton
FR_LR_04	The system should calculate certain individual statistics (hitting percentage, assist percentage, etc.).	Use Cases: 1, 10, 11	1	10/3/18	Joel Walton
FR_LR_05	The system should calculate certain team statistics (blocks per set, digs per set, etc.)	Use Cases:1, 10, 11	1	10/3/18	Joel Walton
FR_LR_06	The system should organize statistics in a dynamic spreadsheet that resembles the NCAA official box score spreadsheet.	Use Cases: 1, 10, 11	1	10/3/18	Joel Walton
FR_LR_07	The system should utilize voice recognition input to allow a single statistician to record statistics orally.	Use Cases: 4, 5, 8	1	10/3/18	Joel Walton
FR_LR_08	The system should allow statisticians to record statistics manually.	Use Case 8		10/3/18	Joel Walton
FR_LR_09	The system should allow coaches and players to generate and view	Use Cases: 1, 6, 10, 11	1	10/3/18	Joel Walton

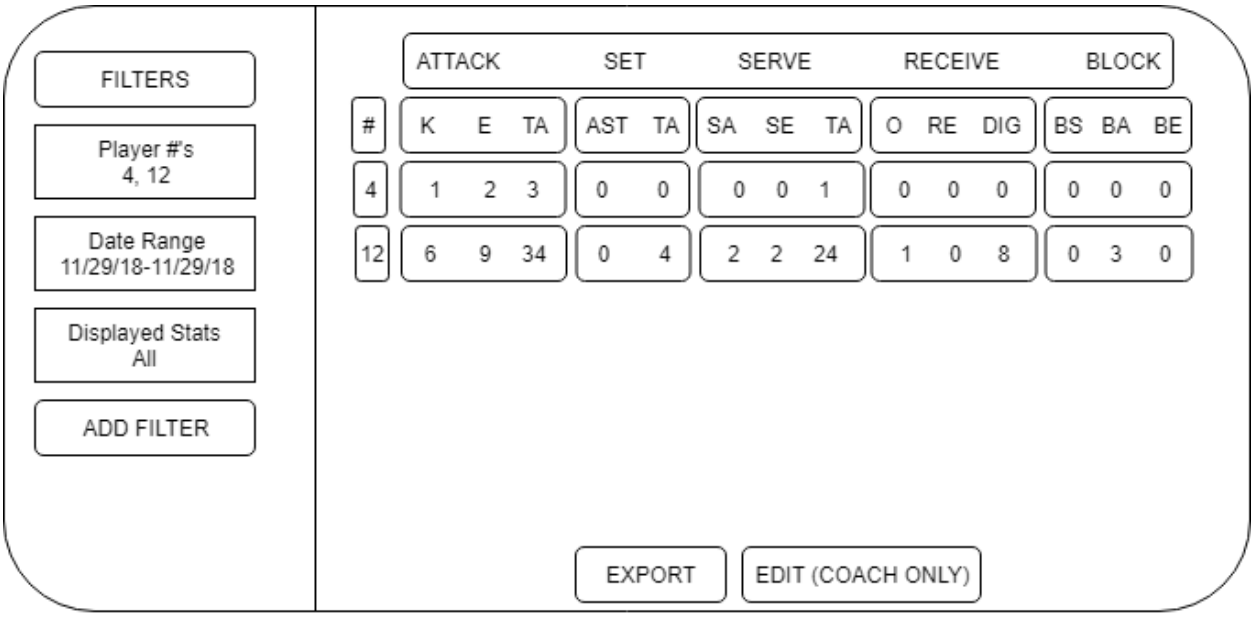
	customized statistics spreadsheets.				
FR_LR_10	The system should allow coaches to choose specific statistics to be tracked from the list of raw statistics.	Use Case 7	1	10/3/18	Joel Walton
FR_LR_11	The system should incorporate the current date and players in attendance when setting up new practices.	Use Case 4	1	10/3/18	Joel Walton

User Interface Requirements:

The following UI mockup is the Single Practice Page where voice/manual statistics will be kept by the statistician. (Use Case 8)



The following UI mockup is the Check Stats Page where players and coaches will go to review stored statistics and generate Excel spreadsheets. (Use Cases 1, 10, 11)



Usability Requirements:

Req#	Requirement	Comments	Priority	Date Rvwd	Client Reviewed / Approved
UR_LR_01	The system should provide visual feedback in response to each voice command	Use Case 8	2	10/3/18	Joel Walton
UR_LR_02	The user documentation should be complete		1	10/3/18	Joel Walton
UR_LR_03	An internet connection is not required to use the system		1	10/3/18	Joel Walton

Performance Requirements:

- Capacity Requirements:

Req#	Requirement	Comments	Priority	Date Rvwd	Client Reviewed / Approved
CR_LR_01	The system should support a maximum team capacity of 100 teams		3	10/3/18	Joel Walton

CR_LR_02	The system should support a maximum of 100 players/team		3	10/3/18	Joel Walton
CR_LR_03	The system should support 20 or more simultaneous users.		3	10/3/18	Joel Walton

- Availability Requirements:

Req#	**Requirement**	**Comments**	**Priority**	**Date Rvwd**	**Client Reviewed / Approved**
AR_LR_01	The system must be available for use during all team practices	Use Case 4	1	10/3/18	Joel Walton
AR_LR_02	Voice recognition must have <5% failure rate	Use Case 8	2	10/3/18	Joel Walton
AR_LR_03	Future iterations must be compatible with previously created teams		3	10/3/18	Joel Walton

- Latency Requirements:

Req#	**Requirement**	**Comments**	**Priority**	**Date Rvwd**	**Client Reviewed / Approved**
LR_LR_01	The maximum acceptable time for system startup should be 5 seconds		3	10/3/18	Joel Walton
LR_LR_02	On average, the delay from voice input to system response should be less than a second	Use Case 8	2	10/3/18	Joel Walton
LR_LR_03	85% of output spreadsheets shall be generated in less than 2 seconds.	Use Case 1	3	10/3/18	Joel Walton

Manageability/Maintainability Requirements:

- Monitoring Requirements:

Req#	Requirement	Comments	Priority	Date Rvwd	Client Reviewed / Approved
MR_LR_01	The system should notify a user when the voice-recognition process fails to recognize a command.	Use Case 8	2	10/3/18	Joel Walton
MR_LR_02	The system should allow for manual error correction of statistics.	Use Case 9	1	10/3/18	Joel Walton

- Maintenance Requirements:

Req#	Requirement	Comments	Priority	Date Rvwd	Client Reviewed / Approved
MR_TR_01	One-character variable names should only be used in loops		3	10/3/18	Joel Walton
MR_TR_02	Variable names should be written in lowerCamelCase		3	10/3/18	Joel Walton
MR_TR_03	Method names should be written in camelCase		3	10/3/18	Joel Walton
MR_TR_04	Class names should be written in UpperCamelCase		3	10/3/18	Joel Walton
MR_TR_05	Constants should be written in uppercase characters separated by underscores		3	10/3/18	Joel Walton

- Operations Requirements:

Req#	Requirement	Comments	Priority	Date Rvwd	Client Reviewed / Approved
OR_TR_01	The system must allow the use of a microphone in order to record stats orally	Use Case 8	1	10/3/18	Joel Walton
OR_TR_02	The system must allow the user to establish the initial team configuration	Use Case 2	1	10/3/18	Joel Walton

OR_TR_03	The system must allow the user to establish the initial practice configuration	Use Case 4	1	10/3/18	Joel Walton
OR_TR_04	The system should afford the user the ability to undo the most recent change when recording stats orally	Use Case 9	1	10/3/18	Joel Walton

Network and Hardware Interfaces:

Currently, PracStat is designed to work only on Windows PCs. The user of PracStat will interface with the system both through voice input and also through hardware such as the computer mouse and keyboard. A touch interface is not required at this time, because PracStat is only designed to work with Windows desktop and laptop PCs. Laptops with touch screens will not require any additional interfaces in order to utilize the touch feature, it will just function as an alternative to using a mouse.

Systems Interfaces:

PracStat will generate an Excel Spreadsheet that the computer will store on the hard drive. After the file has been generated and saved, a user can open the file in Excel or any other compatible spreadsheet viewer.

File Structure and Format:

- The spreadsheet will be a .xlsx file
- The .xlsx file will contain statistics data for each player on a team

File Description:

- The .xlsx file will contain the following data:
    - Player name
    - Player number
    - Statistics (chosen by the coach prior to file generation)

Communications Interfaces:

There will need to be a communication interface between the microphone and PracStat so that the system can accept speech input. This will primarily be handled when the user sets up their microphone and ensures that they have speech services enabled in their computer settings. As long as the microphone is enabled and working, PracStat will be able to process voice input via the microphone.

Security Requirements:

- Protection Requirements:

Req#	Requirement	Comments	Priority	Date Rvwd	Client Reviewed / Approved
SR_PR_01	The system can password lock some features	Stat entry, stat editing, team editing, team creation	1	10/3/18	Joel Walton
SR_PR_02	The system can revert to a backup if needed	Only accessible by coach	3	10/3/18	Joel Walton
SR_PR_03	Spreadsheets generated by players will be read-only	Use Case 1	3	10/3/18	Joel Walton
SR_PR_04	Passwords will be encrypted	Encryption method TBD	3	10/3/18	Joel Walton

- Authorization and Authentication Requirements:

Req#	Requirement	Comments	Priority	Date Rvwd	Client Reviewed / Approved
SR_AR_01	When creating a team, the coach can set passwords for each tier of user	Coach > Statistician > Player	2	10/3/18	Joel Walton
SR_AR_02	Coach can change passwords if needed		3	10/3/18	Joel Walton

Data Management Requirements:

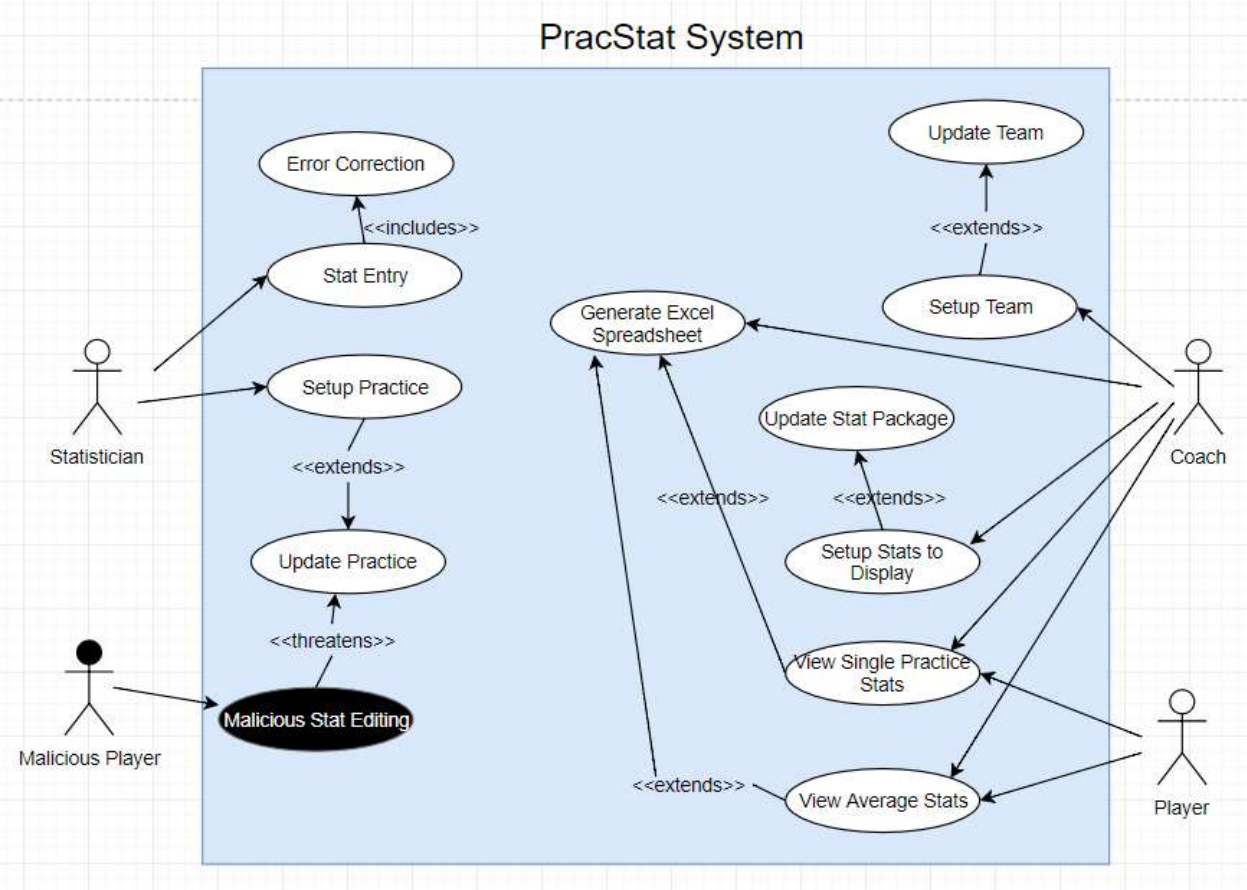
Req#	Requirement	Comments	Priority	Date Rvwd	Client Reviewed / Approved
DR_TR_01	All stored statistics should be accessible for any team member		2	10/3/18	Joel Walton

DR_TR_02	Entered data should be integers		3	10/3/18	Joel Walton
DR_TR_03	Calculated data should be floats		3	10/3/18	Joel Walton
DR_TR_04	Team and practice stats should be stored in a .xlsx file		2	10/3/18	Joel Walton
DR_TR_05	Stats will be initialized to 0 at the beginning of each practice	Use Case 4	1	10/3/18	Joel Walton
DR_TR_06	Once a practice is submitted, the data is read-only for all users except coach		3	10/3/18	Joel Walton

User Scenarios/Use Cases
Actor Profiles:

- Statistician
  - Setup Practice
  - Update Practice
  - Stat Entry
  - Error Correction
- Player
  - View Single Practice Stats
  - View Average Stats
- Coach
  - Setup Team
  - Update Team
  - Setup Stats to Display
  - Update Stat Package
  - Generate Excel Spreadsheet

Use Case Diagram:



Use Cases:

Use Case 1
Generate Excel spreadsheet
Actors
Statistician, Coach
Use Case Overview
After the stats have been entered, either the coach or a statistician wishes to generate an Excel spreadsheet with the data.
Trigger
Stats need to be viewed in an Excel spreadsheet.
Preconditions

- The stats data has already been recorded
- Either the coach or a statistician wants to see the stats in an Excel spreadsheet

Basic Flow: Generate spreadsheet
Description
This scenario describes the situation where the statistician or coach generate an Excel spreadsheet containing the relevant stats. This is the main success scenario.

1. User starts PracStat application
2. User selects relevant team
3. User selects practice/practices containing the data they want
4. User specifies which stats they want in the spreadsheet
5. User selects "generate spreadsheet" option
6. User chooses where to save the spreadsheet file
7. User chooses a name for the file
8. The spreadsheet is saved

Termination Outcome
An Excel spreadsheet is created that contains the chosen stats.
Subflow 4A: Specifying stats to display
Description
This subflow describes in more detail how a user would go about selecting which stats they want to include in the spreadsheet.

4A1 User is shown a box containing all available stats that can be chosen
4A2 User indicates choices by putting a checkmark next to the desired stats

Termination Outcome
Stats have been chosen.
Post Conditions
All selected stats appear in the generated Excel spreadsheet file.
Business Rules

- Only one spreadsheet can be made at a time

Use Case 2
Setup Team
Actor
Coach
Use Case Overview
The coach configures the team for the first time in order to record stats for it.
Trigger
The team has not been added to PracStat and needs to be added to record stats.
Preconditions

- The team does not already exist in PracStat.
- Stats need to be recorded for the team.

Basic Flow: Team Configuration
Description
This scenario describes the situation where the coach configures the team to be displayed and used within PracStat. This is the main success scenario.

1. Coach starts PracStat application
2. Coach selects "Add team"
3. Coach enters team name
4. Coach enters player data
5. Coach confirms team settings
6. PracStat displays team data

Termination Outcome
The team has been successfully added to the system and is currently displayed to the user.

Alternate Flow 3A: Duplicate team name
Description
This scenario describes the situation where a team name already exists and the coach attempts to create another team with the same name.

3A1 Coach is prompted with duplicate team notification
3A2 Coach selects unique team name

Termination Outcome
The coach selects a team name that has not been previously added to the system.
Subflow 4A: Entering player data
Description
This subflow describes in more detail the data that is stored for each member of the team.

4A1 Coach enters player name
4A2 Coach enters player number
4A3 Coach enters player position

Termination Outcome
Data for each team member has been entered into the system.
Post Conditions
All team data is stored and the system is ready to record stats for players of the team.
Business Rules

- Only the coach can create a new team
- Each team must have a unique name
- Each player must be assigned a name, number, and a position

Use Case 3
Update Team Data
Actor
Coach
Use Case Overview
There has been a change within the team that needs to be reflected in the PracStat system. The coach edits the team to modify/add the appropriate data.
Trigger
The team data needs to be updated.
Preconditions

- The current team data is outdated and needs to be updated
- The coach has access to any updated data that needs to be added

Basic Flow: Modify Team Data
Description
This scenario describes the situation where the coach needs to change a setting with the team or with a player on the team. This is the main success scenario.

1. Coach starts PracStat application
2. Coach selects “Edit team”
3. Coach selects data to modify/add
4. Coach makes any needed changes to team data
5. Coach confirms team settings
6. PracStat displays updated team data

Termination Outcome

Any necessary changes to the team data have been implemented.
Post Conditions
All team data is up to date and is displayed to the user.
Business Rules
- Only the coach can modify team data

Use Case 4
Setup Practice
Actor
Coach
Use Case Overview
There is at least one upcoming practice that needs to be specified for a team. The coach enters practice data to ensure that stats can be recorded.
Trigger
The current practice has not been added to PracStat.
Preconditions
- The practice does not already exist in PracStat.
- Stats need to be recorded for the practice.

Basic Flow: Practice Configuration
Description
This scenario describes the situation where the coach configures a practice that is associated with a specific team. This is the main success scenario.
1. Coach starts PracStat application
2. Coach selects appropriate team
3. Coach selects "Add practice"
4. Coach enters practice data
5. Coach confirms practice settings

Termination Outcome
Each practice has been successfully added and is shown in the list of practices for the currently selected team.
Subflow 4A: Entering practice data
Description
This subflow describes in more detail the data that is stored for a practice.
4A1 Coach enters practice name
4A2 Coach enters date of practice
Termination Outcome
Data for the practice has been entered into the system.
Post Conditions
All required practices are stored and the system is ready to record stats for players in each practice.
Business Rules
- Only the coach can configure a new practice
- Each practice must be assigned a name and date.

Use Case 5
Update Single Practice Info
Actor
Statistician
Use Case Overview
The statistician has just finished setting up a practice (described in use case 4) but then realizes that he accidently entered a player's number wrong. He then needs to update this information.
Trigger
The statistician needs to update/change/add information to a practice in PracStat.
Preconditions

- Use case 4 (setup practice) has been successfully completed.

Basic Flow: Update Practice
Description
This scenario describes the situation where the statistician updates practice information in the PracStat application. This is the main success scenario.

1. The statistician selects a single practice date out of a list of practices that has previously been set up.
2. The system displays an "Update Practice" window that allows changes to be made to specific types of information.
3. The statistician updates information as necessary.
4. The statistician clicks a button that saves the changes.
5. The statistician exits the "Update Practice" window.

Termination Outcome
The statistician successfully updated/changed/added information to a single practice in PracStat.
Alternative Flow 5A: The "Update Practice" window is not functioning correctly. Nothing happens when button is clicked.
Description
This scenario describes the situation where the "Update Practice" button is not functioning correctly such as when nothing happens when the statistician clicks the button.

5A1 The statistician deletes the entire practice.
5A2 The statistician the all previous information into a new practice, including the updates.

Termination Outcome
The statistician now has all correct/updated information for the single practice he/she is currently working on.
Post Conditions
The statistician now has all correct/updated information for the single practice he/she is currently working on.

Use Case 6
Set Up Statistics to Display
Actor
Statistician or Coach
Use Case Overview

The coach decides that he would like the output excel spreadsheet to include a new statistic that wasn't included in previous spreadsheets. The statistician or coach change the PracStat settings to include this new statistic.

Trigger

The coach decides that he would like a new statistic to be included in the output excel spreadsheet.

Preconditions

- Initial app setup and team setup has been successfully completed.

Basic Flow: Set Up Statistics to Display

Description

This scenario describes the situation where the coach or statistician change/add which statistics will show up in the output spreadsheet.

1. The statistician or coach selects the "Settings" button on the main application screen and then selects "Spreadsheet Setup".
2. Under the "Displayed statistics:" category, the statistician or coach can check or uncheck which stats he wants displayed from the list of all possible statistics.
3. The statistician or coach clicks the "Save Settings" button and exits back to the main application screen.

Termination Outcome

The coach or statistician successfully changed/updated which statistics will display in the output spreadsheet.

Alternative Flow 6A: Desired Statistic Not In List

Description

This scenario describes the situation where the list of all possible statistics in the "Displayed statistics:" category of the "Settings" page does not list a desired statistic.

6A1 The statistician or coach follow the basic flow of use case 7: Update Stat Package.

Termination Outcome

All desired statistics are listed on the output spreadsheet.

Post Conditions

All desired statistics are listed on the output spreadsheet.

Use Case 7

Update Stat Package

Actor

Statistician or Coach

Use Case Overview

The coach decides that he would like to keep/track a new statistic that is not currently being tracked and is not listed in Settings.

Trigger

The coach decides that he would like a to track a new statistic not currently listed in Settings.

Preconditions

- Initial application team setup has been successfully completed.

Basic Flow: Update Stat Package

Description

This scenario describes the situation where the coach updates the stat package in order to track a new statistic.

1. The statistician or coach selects the “Settings” button on the main application screen and then selects “Spreadsheet Setup”.
2. Under the “Update Stat Package” category, the statistician or coach can check or uncheck which statistics are able to be tracked.
3. The statistician or coach clicks the “Save Settings” button and exits back to the main application screen.

Termination Outcome
The coach or statistician successfully added a new statistic to track.
Post Conditions
All desired statistics are being tracked for each practice.

Use Case 8
Stat Entry
Actor
Statistician
Use Case Overview
A new practice scrimmage is about to begin and coach wants the statistician to track statistics using the PracStat application using voice so that they are as accurate as possible.
Trigger
A new practice is about to begin and the coach wants the stats to be tracked using PracStat.
Preconditions

- Initial app setup, team setup, and practice setup have all been successfully completed.

Basic Flow: Voice Stat Entry
Description
This scenario describes the situation where the statistician tracks statistics using voice input.

1. In the single practice screen, the statistician selects the “Voice Stat Entry Mode” button.
2. In this mode, the statistician clicks and holds down the “Record” button while he speaks into the microphone.
3. The statistician uses voice to record each player’s individual statistics using the directions included in the user manual.

Alternative Flow 3A: The Voice Recognition Functionality Fails to Record Voice Stats or Statistician Prefers Manual Stat Keeping
Description
The statistician has noticed that the voice recognition functionality for tracking stats is not working properly, or the statistician prefers manual stat keeping.

3A1: The statistician clicks the “+” and “-” buttons that are available in each stat box to add or subtract from a player’s stats

3A2: The statistician no longer needs to use his/her voice

3A3: All other processes of the application remain the same. The statistics were just entered using a different method.

Termination Outcome
The statistician successfully tracked accurate statistics from the practice for the coach or players to review afterward.
Post Conditions
Accurate practice statistics are tracked using PracStat.
Business Rule

- The laptop device that the statistician uses must have a working microphone to capture voice input.

Use Case 9
Error Correction
Actors
Statistician
Use Case Overview
While recording statistics during a practice, the statistician realizes he has made a mistake on the previous call. He selects the undo button and attempts to record the play correctly.
Trigger
A PracStat user wants to delete or change a recorded stat
Preconditions
- There is an ongoing practice
- The statistician has made an error in recording/The voice recognition made an error

Basic Flow: Error Correction
1. The user notices that the stat that has just been recorded is incorrect
2. The user presses the undo button
3. The system deletes the previously recorded input
4. The user inputs the new corrected statistic

Alternative Flow 4A: The Undo Button Was Pressed by Mistake
Description
The user has noticed that the correct stat was recorded, but deleted by the undo button
4A1: The user notes that they had already recorded the correct stat
4A2: The user selects a redo button
4A3: The system restores the previously deleted stat
Termination Outcome
The correct stat has been entered into the system
Business Rules
Due to the nature of undo buttons, if multiple stats have been recorded since the incorrect stat, all stats following the erroneous stat will have to be input a second time

Use Case 10
User Views Single Practice Stat.
Actors
Coach/Player
Use Case Overview
After practice, a team member wishes to view his statistics from the day's practice. The player opens the application, navigates to the collection of stats containing information about the specific day, then uses a filter to show only his individual statistics.
Trigger
Player wishes to check his statistics from the last practice
Preconditions
- Information has been collected from one or multiple team practices.
- Players have been taught to navigate the program.

Basic Flow: Check Single Practice Stats

Description
This scenario describes the situation where a team member looks for previously entered stats from a specific practice.

1. The system displays its main menu which contains links to teams, practices, and statistics pages for that particular user
2. User navigates to the statistics connected to his team and the day's practice
3. The system presents a spreadsheet containing the information for all players for that particular practice
4. The user enters a jersey number or other identifying information into a filter
5. The system updates the spreadsheet to present all information pertaining to that player for the day's practice

Termination Outcome
The player has seen the stats he wanted to see.
Alternative Flow 5A: The player is not displayed in the filtered statistics
Description
This scenario describes the situation where the user has entered a player's information into the filter, but no information is displayed on the spreadsheet.

5A1 The user verifies that he input the correct information into the filter
5A2 The user selects a different practice, as the selected practice may not have any information for that player

Termination Outcome
The user locates the data he needs.
Post Conditions
All information the user wants to see has been displayed
Business Rules
The player can only check his statistics on a team tablet or laptop, so the information can only be checked during the hours which they have access to the team devices.

Use Case 11
User Views Average Stats
Actors
Coach/Player
Use Case Overview
After practice, a team member and/or the coach wishes to view his average statistics from any number of past practices. The user opens the application, navigates to the team's overall statistics, uses a filter to show statistics from the desired number of past practices, then uses a filter to show only one player's individual statistics.
Trigger
Player wishes to check his average statistics from multiple previous practices
Preconditions

- Information has been collected from multiple team practices.
- Players/Coaches have been taught to navigate the program.

Basic Flow: Check Average Stats
Description
This scenario describes the situation where a team member looks for a specific group of previously entered stats containing data from multiple practices.

1. The system displays a menu and asks for user input
2. Player navigates to the statistics connected to his team's overall statistics
3. The system presents a spreadsheet containing the information for all players for all recorded practices.
4. The user enters a jersey number or other identifying information into a filter
5. The user uses another filter to select the range of practices he wishes to include in the displayed data
6. The system calculates averages and updates the spreadsheet to present all information pertaining to that player for the selected range of practices

Termination Outcome

The player has seen the stats he wanted to see

Alternative Flow 6A: No Data in the Selected Range

Description

This scenario describes the situation where the user has entered a range of dates, but no information is displayed on the spreadsheet.

6A1 The user verifies that he input the correct information into the filter

6A2 The user selects a different date range, as there may not be any recorded practices for the selected dates

Termination Outcome

The user selects a range of dates that contain practice data

Post Conditions

All information the user wants to see has been displayed

Business Rules

The player can only check his statistics on a team tablet or laptop, so the information can only be checked during the hours which they have access to the team devices.

Misuse Case 1

Name: Malicious stat editing

Summary: A player maliciously edits their stats to either make themself look better or make someone else look bad.

Basic Path:

BP1 The player gains access to the coach's password

BP2 The player logs in as the coach with coach privileges

BP3 The player loads a past practice session

BP4 The player maliciously edits the stats without the knowledge of the coach

Alternate Paths:

AP1 The coach leaves his computer while still logged in to the PracStat application

AP2 The player gains access to the coach's computer

AP3 The player has the coach's privileges

AP4 The player loads a past practice session

AP5 The player maliciously edits the stats without the knowledge of the coach

Mitigation Points:

MP1 Password does not work - Coach already logged in (BP2)

MP2 Password does not work - Changed (BP2)

Preconditions:

Pc1 Operator has special authority

Pc2 Spreadsheets are editable by users with appropriate privileges

Assumptions:

As1 Player uses a password to enter the application (for bp2)

As2 Coach's computer does not timeout (for ap1)

Worst Case Threat:

Wc1 The malicious player gets coach privileges

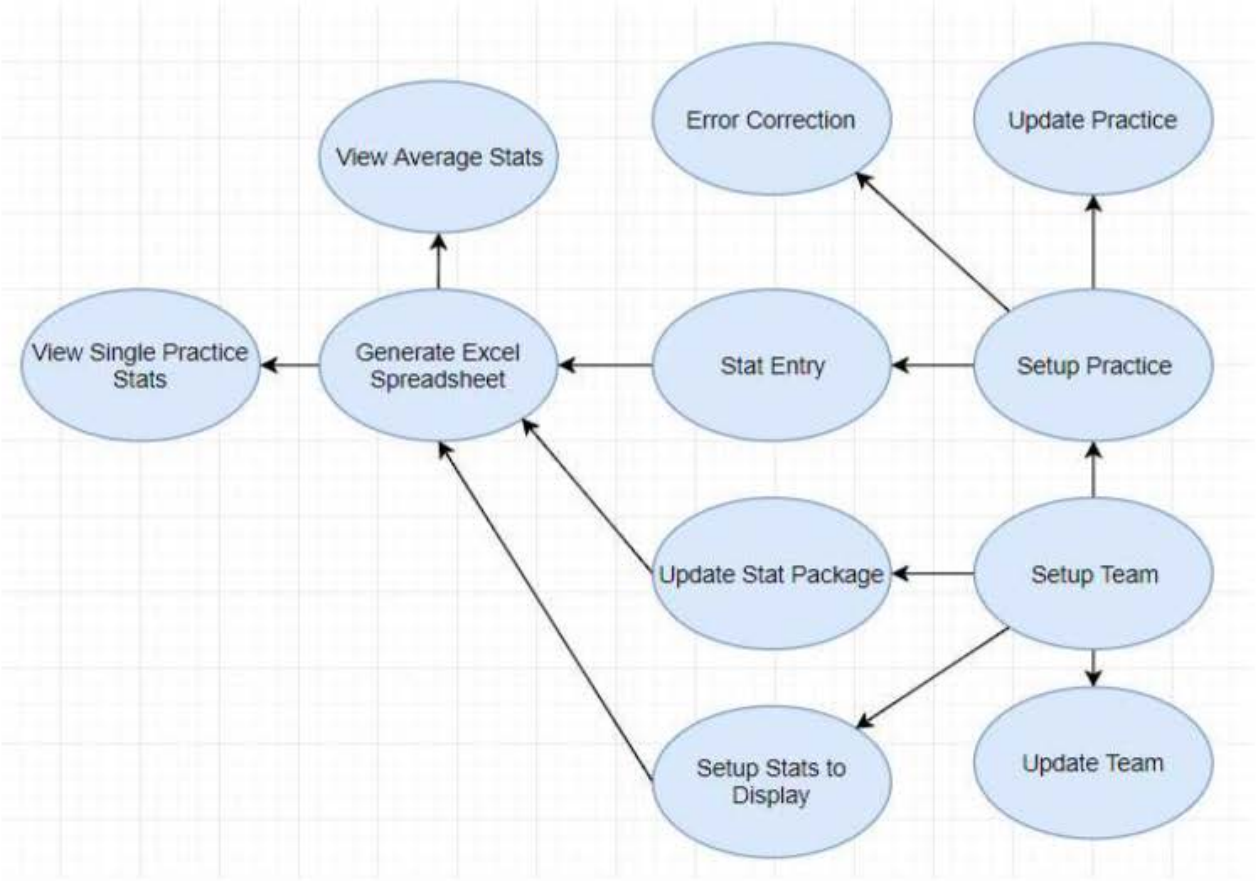
Use Case Ranking and Priority Matrix:

Evaluates use cases on 1-5 scale against six criteria.

1. Significant impact on the architectural design.
2. Easy to implement but contains significant functionality.
3. Includes risky, time-critical, or complex functions.
4. Involves significant research or new or risky technology.
5. Includes primary functions.
6. Will increase revenue or decrease costs.

Use Case Name	Ranking Criteria, 1-5						Total Score	Priority	Build Cycle
Setup Practice	2	4	1	2	4	-	13	MEDIUM	1
Update Practice	3	2	1	2	2	-	10	LOW	2
Stat Entry	5	4	5	5	4	-	23	HIGH	1
Error Correction	2	4	4	3	4	-	17	MEDIUM	2
Setup Team	2	3	1	2	4	-	12	LOW	1
Update Team	2	2	1	2	2	-	9	LOW	2
Update Stat Package	3	2	1	2	2	-	10	LOW	2
Setup Stats to Display	3	3	1	2	4	-	13	MEDIUM	1
View Single Practice Stats	3	4	2	3	4	-	16	MEDIUM	2
View Average Stats	3	4	2	3	4	-	16	MEDIUM	2
Generate Excel Spreadsheet	4	4	1	5	5	-	19	HIGH	1

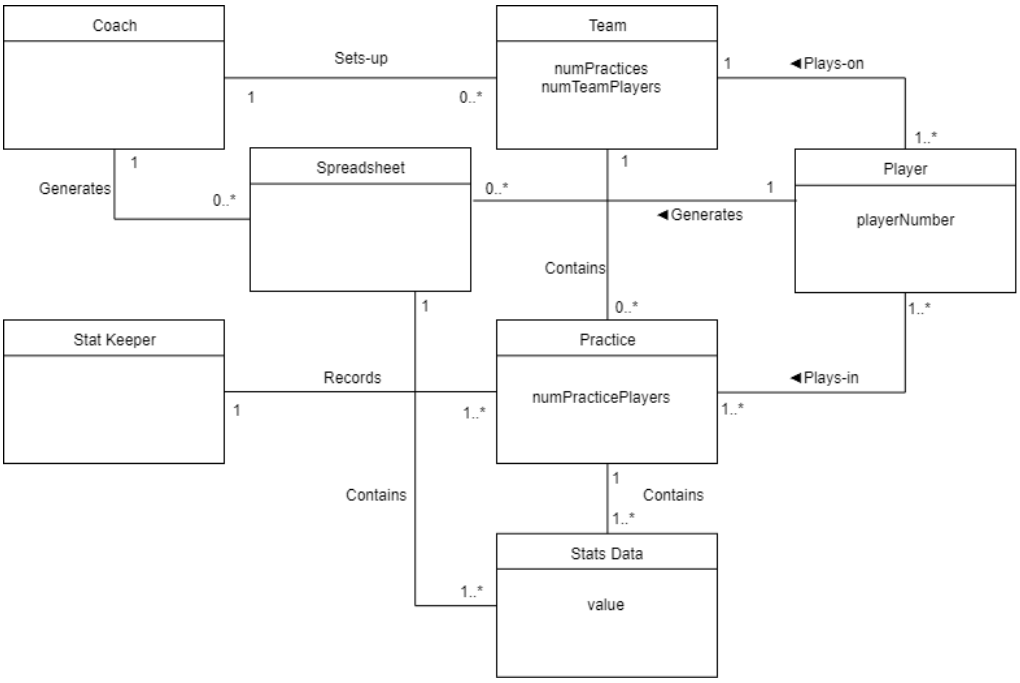
Use Case Dependency Diagram:



Use Case and Requirements Traceability Matrix:

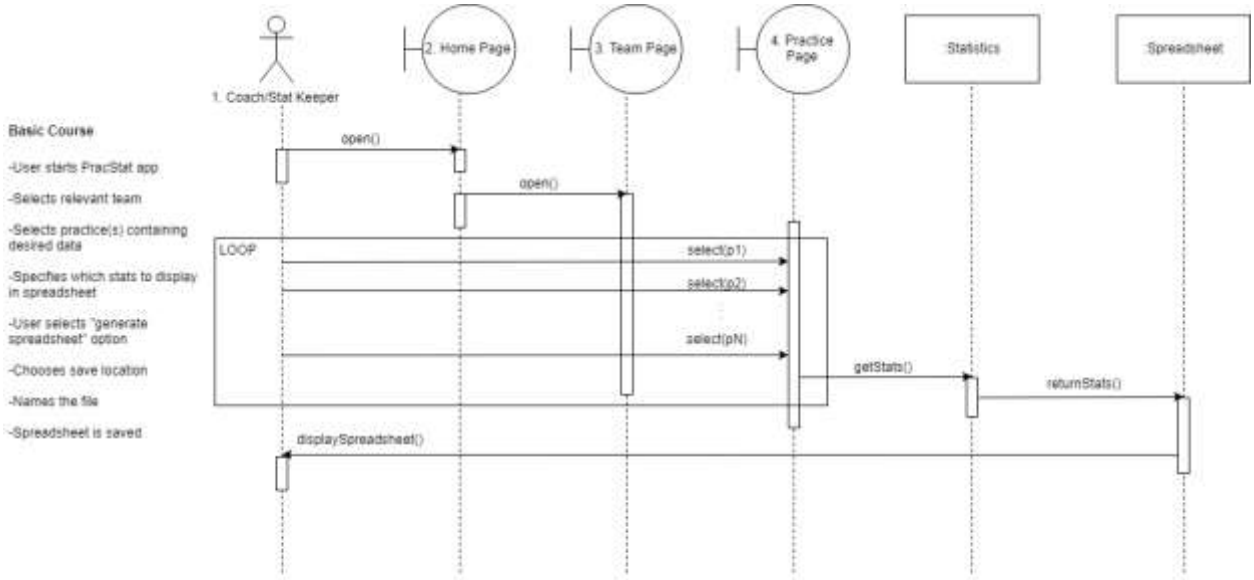
	USE CASES											
REQUIREMENTS	1	2	3	4	5	6	7	8	9	10	11	12
FR_LR_01		X	X									
FR_LR_02	X							X		X	X	
FR_LR_03		X	X									
FR_LR_04	X									X	X	
FR_LR_05	X									X	X	
FR_LR_06	X									X	X	
FR_LR_07				X	X			X				
FR_LR_08								X				
FR_LR_09	X					X				X	X	
FR_LR_10							X					
FR_LR_11				X								
UR_LR_01								X				
AR_LR_01				X								
AR_LR_02								X				
LR_LR_02								X				
LR_LR_03	X											
MR_LR_01								X				
MR_LR_02									X			
OR_TR_01								X				
OR_TR_02		X										
OR_TR_03				X								
OR_TR_04									X			
SR_PR_03	X											
DR_TR_05				X								

Domain Model:

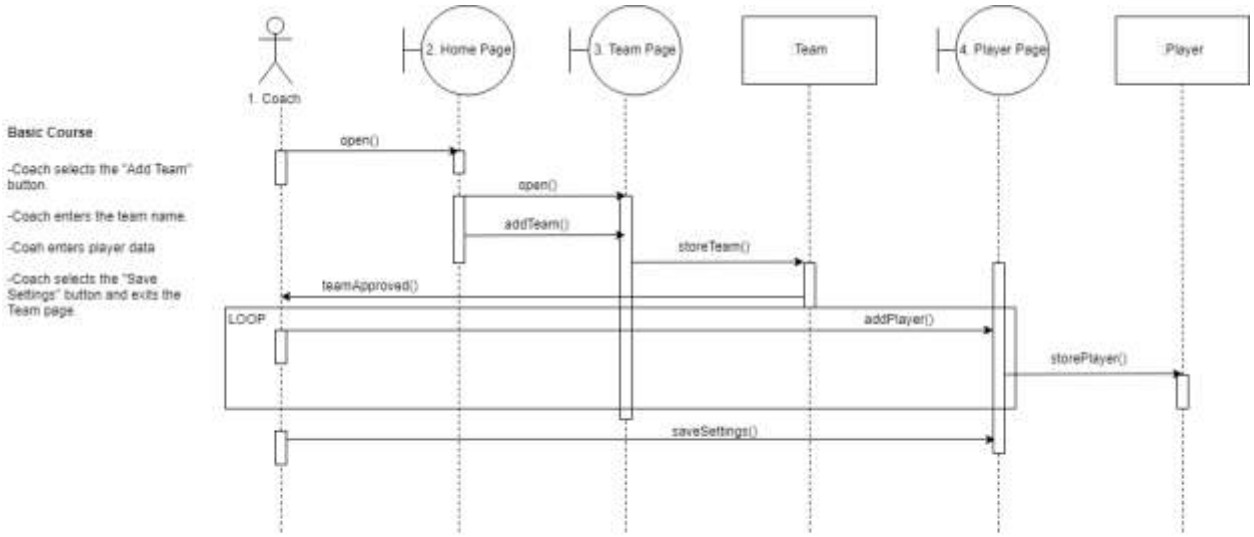


Sequence Diagrams:

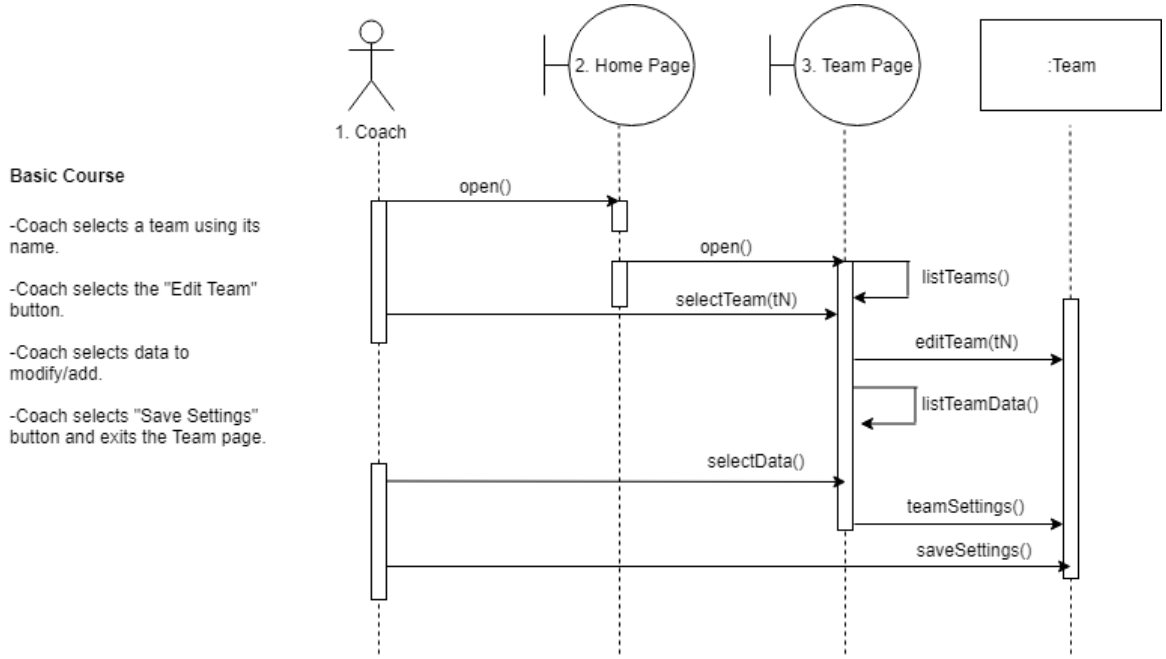
1. Generate Excel Spreadsheet



2. Setup Team



3. Update Team



4. Setup Practice

Basic Course

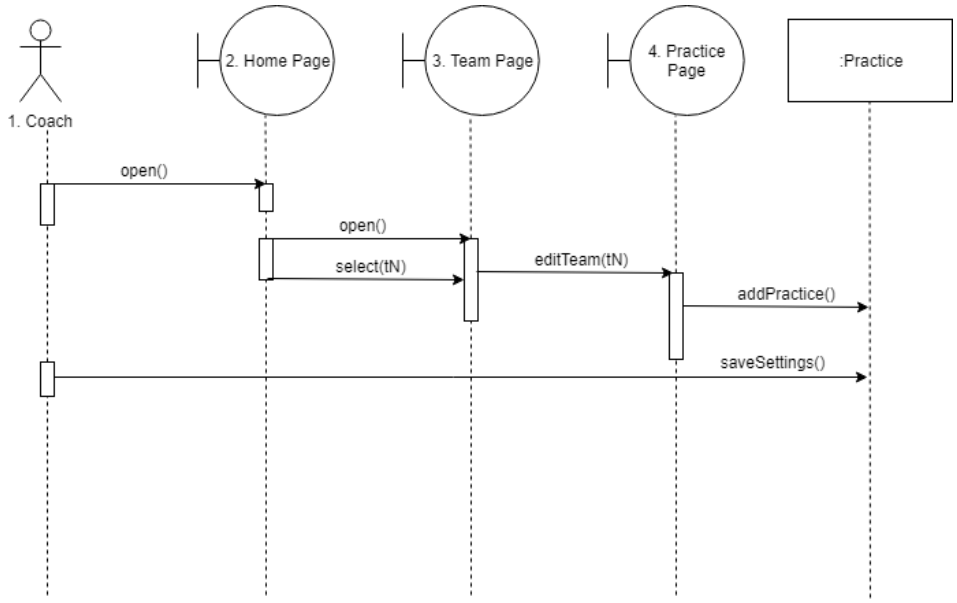
-Coach selects a team by its name.

-Coach selects the "Edit Team" button.

-Coach selects the "Add Practice" button.

-Coach enters practice data.

-Coach selects the "Save Settings" button and exits the Practice page.



5. Update Practice

Basic Course

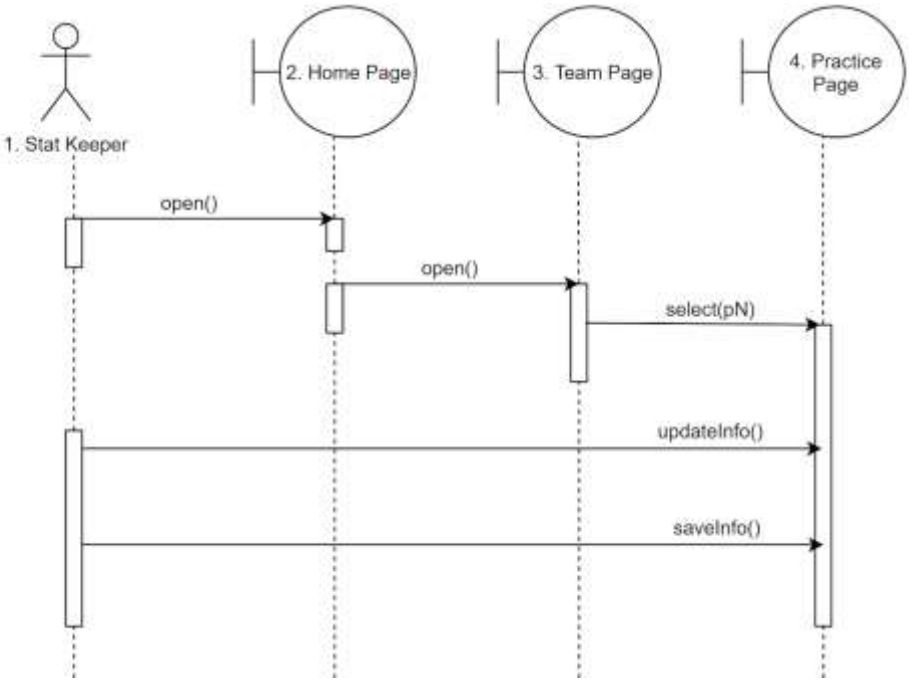
-Stat Keeper selects a single practice date out of practice list.

-System displays "Update Practice" window that allows changes to be made.

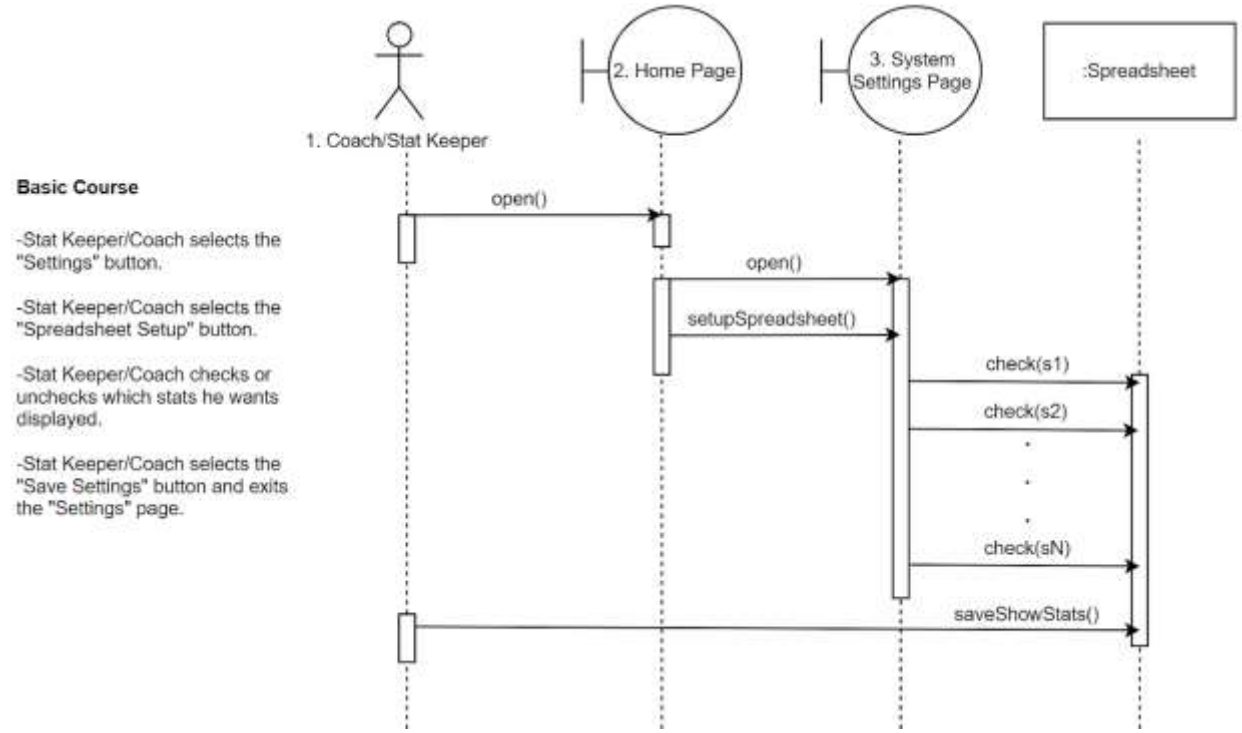
-Stat Keeper updates information as needed.

-Stat Keeper saves changes.

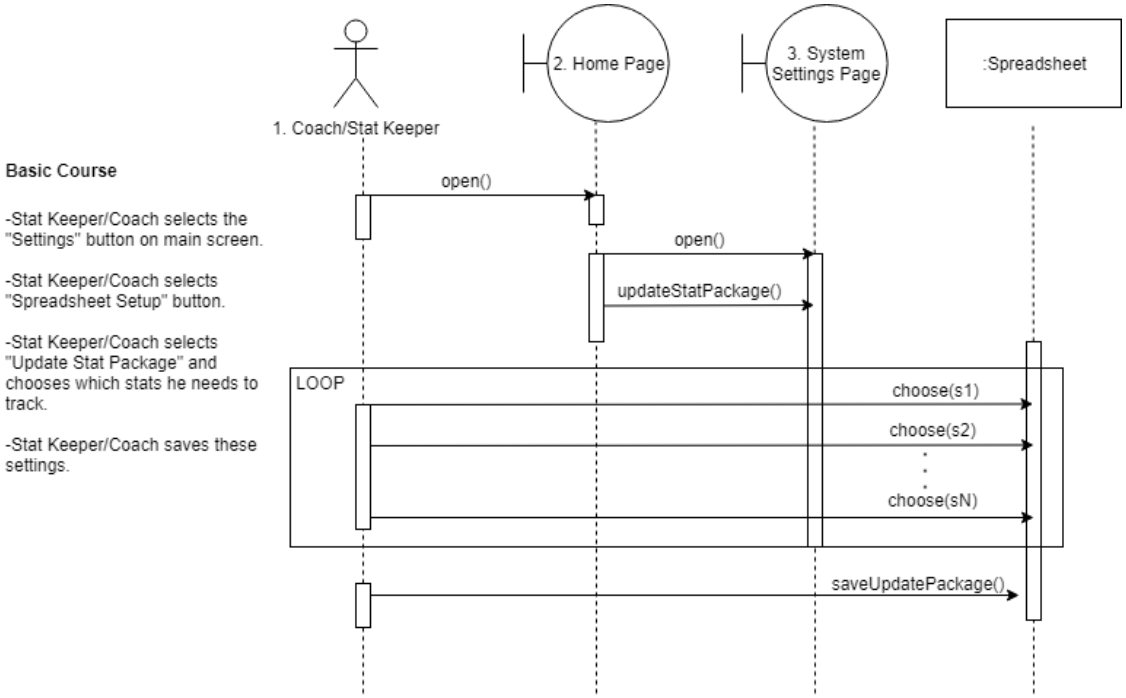
-Stat Keeper exits the "Update Practice" window.



6. Setup Statistics to Display



7. Update Stat Package



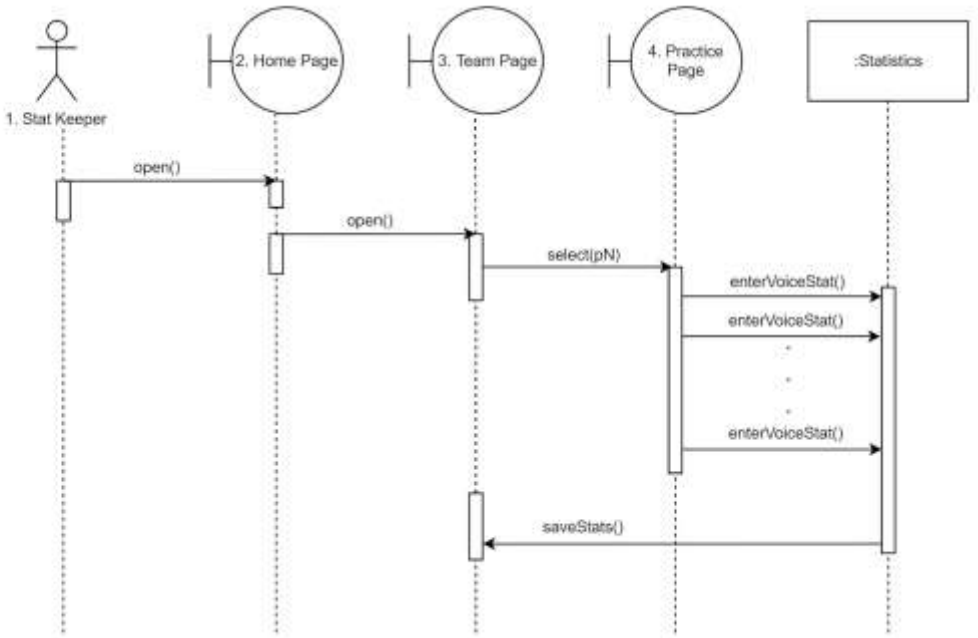
8. Stat Entry

Basic Course

-Stat Keeper navigates to the "Single Practice Screen" and selects the "Voice Stat Entry Mode" button.

-Stat Keeper clicks and holds down the "Record" button while speaking into the microphone.

-Stat Keeper uses voice commands to record each player's individual stats during practice.



9. Error Correction

Basic Course

-User starts PracStat app

-Selects relevant team

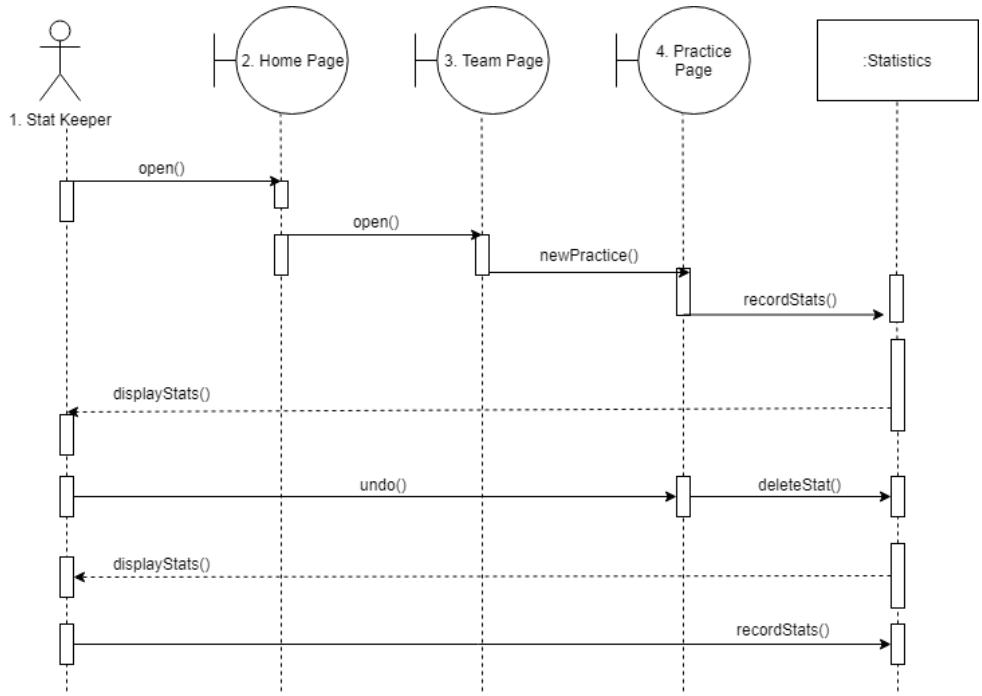
-Sets up new practice

-Records at least 1 stat

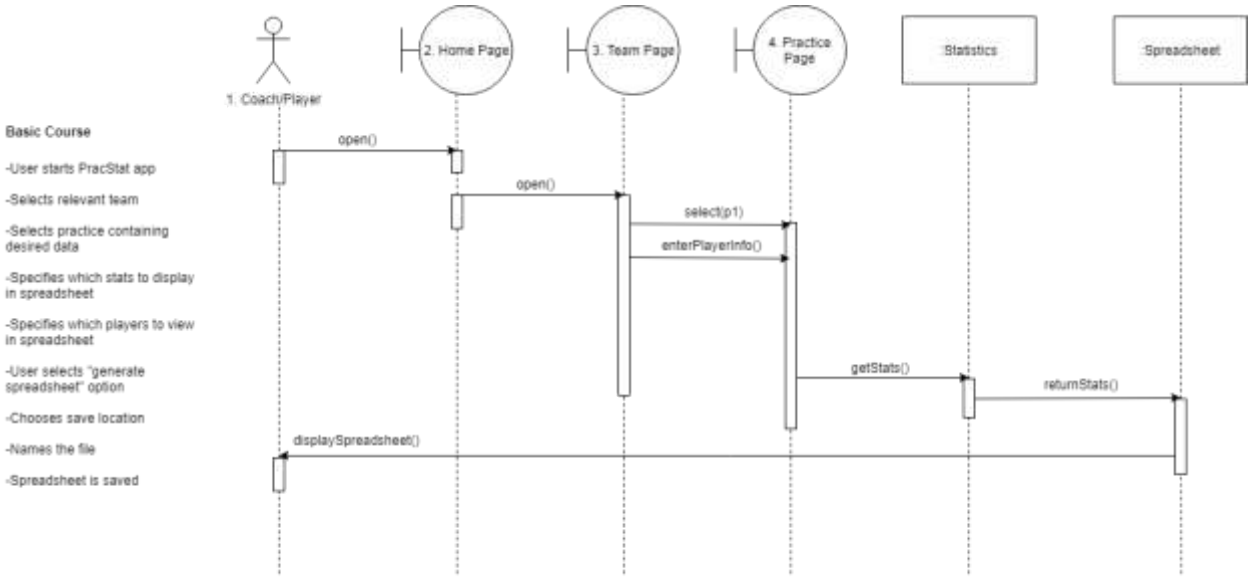
-Realizes stat is incorrect

-Presses undo button

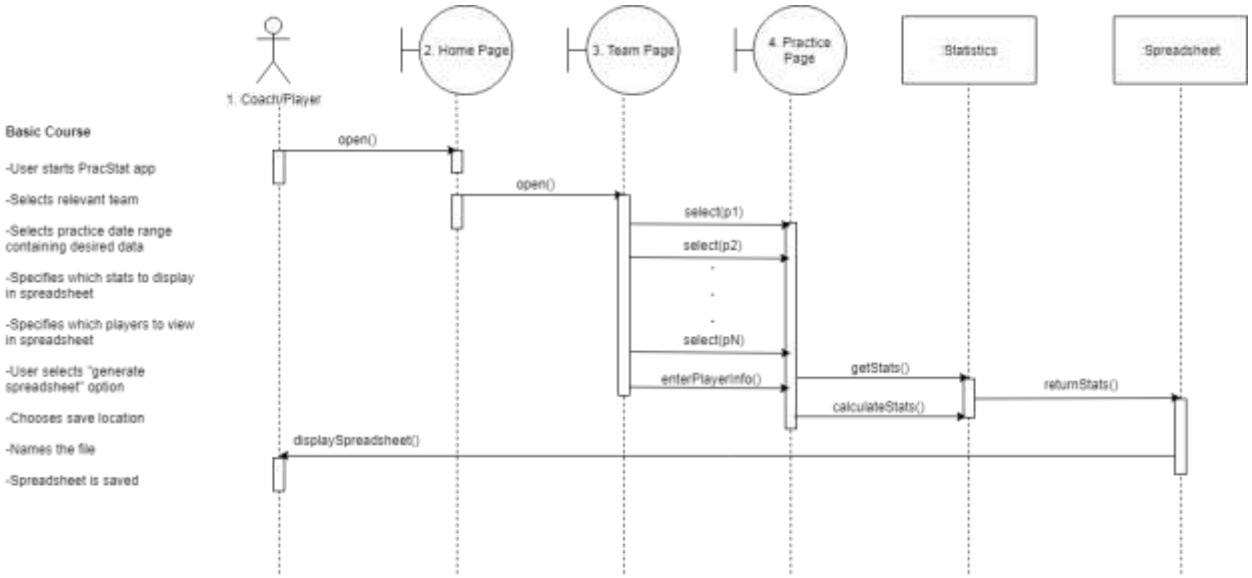
-Records stat for 2nd time



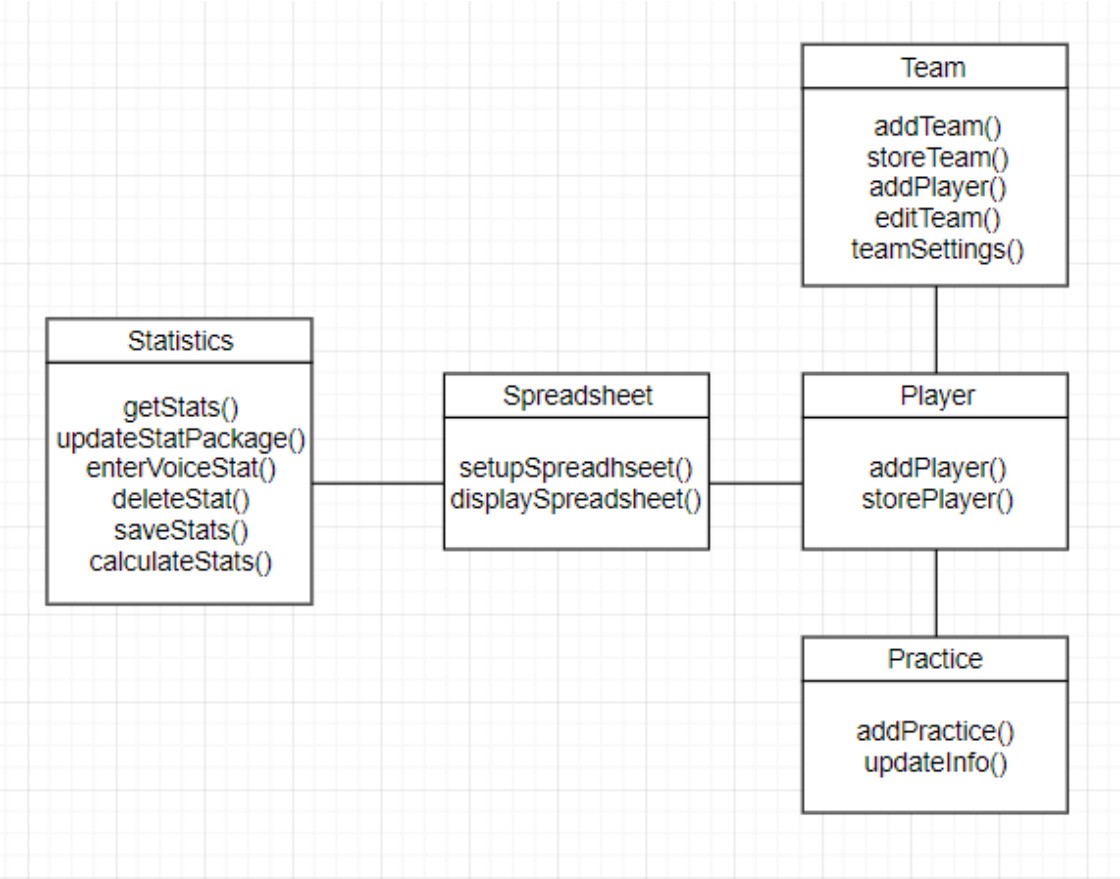
10. View Single Practice Stats



11. View Average Practice Stats



Preliminary Class Diagram:



Deleted or Deferred Requirements

Req#	Business Requirement	Status	Comments	Pri	Date Rvwd	Client Reviewed /Approved
N/A	The system should handle simultaneous practices	Nov 2018: Deleted.		1	10/3/18	N/A
N/A	The system should email the coach any generated spreadsheets	Deferred to future iterations		3	10/3/18	N/A

Requirements Confirmation/Stakeholder Sign-Off

Meeting Date	Attendees (name and role)	Comments
10/3/18	Alex Bobay, Developer Dexter Edwards, Developer Adam Wessel, Developer Joel Walton, BSU Men's Volleyball Head Coach	Confirmed BR_LR_01 – BR_LR_08 Elicited requirements, asked more questions, discussed features
9/5/18	Alex Bobay, Developer Dexter Edwards, Developer Adam Wessel, Developer Joel Walton, BSU Men's Volleyball Head Coach	Proposed project idea to Coach Walton, discussed problems with the current stat-tracking situation, and received Walton's approval to be our client partner.

Quality Assurance Test Plan

Introduction

PracStat will be an application used by volleyball teams at any level to record and analyze player and team stats throughout one or multiple seasons. Statisticians will be able to record stats live during a practice or game using voice or manual input features, and those stats will be saved within the application. After a practice or game has been recorded, the coach and players will be able to generate customizable spreadsheets to display specific statistics from individual sessions, or averages from all or some recorded sessions.

Objectives and Taks

The main objective of the Master Test Plan is to organize all information regarding the QA of our product, PracStat. It will define all tasks to be completed and delegate those who are responsible for each task. The Master Test Plan will also act as a vehicle for communication between PracStat team members, our client, and our professor. It will outline our overall testing process. Additionally, this document is to be used as a service level agreement between the PracStat team and our client - Joel Walton. This SLA will clarify particular aspects of the product - quality, availability, and responsibilities - that will be agreed upon by all PracStat clients.

This Test Plan will include our approaches to testing PracStat both during production and after its release. The main task will be to ensure that PracStat does not have any critical errors that will inhibit its use by the BSU men's volleyball team. Through this plan, we hope to identify problems with our product and fix them in a timely manner. As we will be mostly testing our beta code at the moment, this will be our main focus. From here, we will assess how our code is functioning and make any necessary changes as needed.

Scope

The main areas of testing will include:

- Voice Recognition
- Spreadsheet Generation
- Statistic Calculation
- Configuration
- Number Entry
- Text Entry
- System Response

Tactics

Voice recognition will be tested by having multiple people with a wide range of voices speak into the application and determining if what the application determined was said is equal to what the user actually said. We will test mainly for statistic keywords such as “dig”, “block”, and “kill”, in addition to numbers. The test will also include rejections of invalid words the application will not need to recognize, including non-english words.

Spreadsheet generation will be tested by selecting a specific set of statistics and using the application to generate an excel spreadsheet using those statistics. If the spreadsheet contains all necessary statistics in the correct locations and is in the format of the official NCAA box score, the test passes. If the spreadsheet is missing any of the selected statistics, the statistics are in the wrong locations, or the format does not line up with official NCAA box scores, the test fails.

Statistic calculation testing will be a series of simple tests, as most calculations will be finding averages or totals or previously recorded stat entries. We will first test that any numbers inputted to calculations are positive, as it is impossible to have negative numbers in volleyball statistics. Next, we will test to see if simple functions such as adding or dividing 2 numbers produce an accurate answer, and then combining those to test if the application produces accurate averages. If a correct answer is provided, the test passes. If the answer is negative or incorrect, the test fails.

Configuration will be a simple test, checking to see if a valid PracStat configuration file is present. Upon selecting a file, the test will pass if the file is appropriate XML format and contains one or multiple teams, or fail if the file is invalid or does not exist.

Number entry testing will be setup to only allow certain numbers to pass. PracStat will only be dealing in positive numbers, so any number between 0 and 2,147,483,647 will pass. Blank inputs, numbers outside the established range, a-z and A-Z characters, and special characters will fail this test.

Text entry will be used mainly for player names and will be limited to a-z, A-Z, and 0-9 characters. The text entry testing will pass when presented with any of these characters, and will fail when presented with a blank input or special character. Passwords will be the exception to this test and will allow any ASCII characters.

System response will be tested along with voice input, and tests that the application will notify the user if it fails to detect a voice input. If the user is presented with an error notification

after invalid input is presented to the application, the test passes. If the application does nothing when presented with invalid input, the test fails.

Testing Strategy

The previous section gives a broad overview of the different testing groups that we have identified for this project. Listed below are the different strategies that we plan to use in testing each category of features.

Voice Recognition:

For the voice recognition feature, we plan to primarily use black box testing as voice recognition can lead to a variety of different outcomes as previously mentioned in the tactics section of the scope. There will be key words that we will be testing against the sphinx4 library which will include the number of a player followed by a command such as “Dig”. As we do not need to be actively looking at the code to test the voice recognition, we do not foresee any white box testing that will need to be done for this group.

Spreadsheet Generation:

To test the spreadsheet generation feature, we will use a combination of both black and white box tests. For the black box tests, we will look at special cases that could present problems under certain conditions. One example of this would be a group of stats that do not have anything recorded. Outside of special cases like this, we will also use white box testing to ensure that each type of statistic shows up in the generated CSV file. This will require us to thoroughly test each possible path in the part of our code that will handle the creation of the CSV file using the recorded statistics.

Statistic Calculation:

To calculate the statistics that are not entered in by the user (ex. averages and totals), we will use black box tests to ensure that these calculations are correct. It will not be necessary to use white box tests for this group because we don’t need to know every possible combination of the different calculations that are required. Java’s math class will take care of this and we will just need to make sure that the results of the calculations are correct. We will also need to be mindful of data in different formats in the case of some stats containing decimal points or not.

Configuration:

The initial configuration will be tested using black box testing. For this process, we will look at the result of the parsed XML and confirm that the configuration screen appears if the configuration file does not contain any team data. This is only a one-time process and as a result, will not require nearly as much time as the other groups of features.

Number Entry:

To test the entry of numbers we will use black box tests that handle special cases where the numbers entered are either in range or out of range. In our case, we will not allow any negative numbers and will allow a maximum number of 2,147,483,647. Realistically, no one will ever even come close to hitting the maximum allowed value for a 32-bit integer, so we will mostly be looking at special cases where a user either does not enter something that needs a number, or enters something that is not a number.

Text Entry:

Similar to the process of testing the entry of numbers, text entry will also be tested using black box tests. The main interaction in which a user will need to type in text will be during the configuration of both teams and players. For this, we will only allow letters and numbers to be used. Additionally, we will need a few white box tests to ensure that the program correctly handles cases where duplicate team names exist. This will require us to actively look at our code to determine which actions need to take place given this condition. When setting a password, we will allow the user to include any ASCII character. This will be another black box test included in this feature set.

System Response:

To test the system response, we will focus on white box tests as the response could vary based on the path taken in the code. Our main goal for this section is to ensure that the user is notified whenever an error occurs so that they are not left in the dark. We will be mostly looking at responses to voice recognition errors, but we will also handle other errors that may occur during the use of the program such as duplicate team names.

Unit Testing

Definition:

Unit Testing is the first level of software testing where individual components (units) of a program are tested to validate that each unit performs as designed. At a minimum, unit tests will need to cover 95% of production code in order to be considered adequate. Each unit test will either "pass" or "fail", i.e., no unit test can be partly met. Ideally, each critical method of the system will be paired with a unit test.

Methodology:

Since Java is being used to code PracStat, JUnit will be used to conduct all unit tests. Test scripts will be written simultaneously as production code is written, i.e., each component of the program will be tested as it is written. Throughout development, all unit tests will run each time the PracStat program runs to make sure that new code is not "breaking" previously written code.

System and Integration Testing

Definition:

System and Integration testing will combine many of the lower level tests into a larger test analyzing the overall functionality of the PracStat application. This will be used to test completion of higher level requirements, such as time taken for spreadsheet generation and automatically including team players in new practices.

Methodology:

System and Integration testing will begin once the larger components of PracStat have been completed. The unit testing will include confirmations that the list of players is accessible from the new practice page, the time taken to gather stats and generate a spreadsheet is below 2 seconds, and any other testing that involves multiple system components.

Performance and Stress Testing

Definition:

Performance and Stress Testing checks how the system behaves and performs under specific conditions. Performance Testing examines aspects of the system such as responsiveness, stability, scalability, reliability, speed, and resource usage, while Stress Testing checks the upper limits of the system by testing it under extreme loads.

Methodology:

Performance and Stress Testing will be a very important layer of testing the PracStat application because the responsiveness and reliability of the system (especially the voice recognition feature) is a significant concern. As of now, all performance test scripts and stress test scripts will be written by the PracStat team. Performance testing will mainly be utilized to test the performance of PracStat's voice recognition feature. We will test voice commands of different speeds, accents, slurs, etc. and examine the reliability, responsiveness, and correctness of the information. We will then use this information to try to improve upon the performance of the application.

User Acceptance Testing

Definition:

The purpose of acceptance test is to confirm that the system is ready for operational use. During acceptance test, end-users (customers) of the system compare the system to its initial requirements.

Methodology:

The User Acceptance testing will be conducted primarily with coach Joel Walton as he is the one who will need to sign off on the project in the end. During this time, we will present our project to Joel and evaluate his reactions to how it works. We will all contribute to writing the test scripts. For the sequence of events, we will have a copy of the SRS document with the listed requirements and will have Joel run through each requirement and demonstrate that our program successfully accomplishes them all. In the event that one or more requirements are not met, we will keep notes and try to fix them at a later time.

Automated Regression Testing

Definition:

Regression testing is the selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still works as specified in the requirements.

Methodology:

For the regression testing procedure, we will go back and re run previous tests to ensure that our current code still works as we expect it to. Since our project is relatively small, we do not foresee ourselves devoting a large amount of time to regression testing. However, it will still be important to make sure that any new features that we add do not create any new problems for other code.

Hardware Requirements

Computers: Windows 7 or Windows 10 Operating System

Microphone: A microphone with input capability is required

Environment Requirements

Main Frame:

As PracStat is intended to work only on Windows, the testing environment will need to contain a Windows system, preferably Windows 7 or Windows 10. Unit testing will be done in IntelliJ using JUnit, higher level tests such as the System and Integration Testing will be done in the stand-alone PracStat application. A keyboard will be required for setting up a team and testing number/text entry. For testing voice input, the hardware will require a functional microphone either built in or connected via USB. PracStat is meant to be used in potentially noisy environments, so some testing will need to be done with the addition of a considerable amount of background noise.

Test Schedule

3/21/19 - Complete Unit Testing
3/31/19 - Complete System and Integration Testing
4/14/19 - Complete Performance and Stress Testing
4/21/19 - Complete User Acceptance Testing
4/28/19 - Complete Automated Regression Testing

Control Procedures

Problem Reporting:

For the task of reporting problems, we will keep track of any incidents that occur on an individual basis. With this we will note the date, time, and what we did that caused the error. Ideally, we will include the exact spot in the code where the problem originated, but this may not always be possible. In the event that this is not possible, we will keep track of what we were working on and what lead to the issue. Depending on the seriousness of the problem, we will also rank how high on the list of priorities it is for us to resolve. This will allow us to concentrate our time on resolving the most critical issues first.

Change Requests:

Any modifications that are to be done to the software that are not included within the requirements set forth by the SRS document should be approved by all PracStat members. As time is the limiting factor in this project, we may not be able to successfully complete many outside changes that arise outside the scope of the SRS document. Upon approval, we will implement new features which will be committed to version control to ensure that they work with our already-implemented features. This will help us identify any modules which would be affected by future changes and would allow us to revert back to a prior version if necessary.

Features to be Tested

- Voice Recognition
- Spreadsheet Generation
- Statistic Calculation
- Configuration
- Number Entry

- Text Entry
- System Response
- System Response Time
- Spreadsheet Generation Time
- Manual Stat Entry

Features Not to be Tested

The appearance/display of the PracStat application will not be formally tested; however, decisions regarding the appearance of specific windows, screens, etc. will be carefully considered. An informal kind of acceptance testing will be performed in order to validate the aesthetic of the PracStat application.

Resources/Roles and Responsibilities

Alex Bobay (Developer) Unit Testing, System and Integration Testing, Performance and Stress Testing, Automated Regression Testing

Adam Wessel (Developer) Unit Testing, System and Integration Testing, Performance and Stress Testing, Automated Regression Testing

Dexter Edwards (Developer) Unit Testing, System and Integration Testing, Performance and Stress Testing, Automated Regression Testing

Joel Walton (User) Provides testing environment, User Acceptance Testing

Dolores Zage (User) User Acceptance Testing

Schedules

Major Deliverables:

3/19/19 - Equivalence Partitioning Presentation
3/21/19 - Test Plan
4/28/19 - Test Cases
4/28/19 - Test Incident Reports
4/28/19 - Test Summary Reports

Significantly Impacted Departments (SIDs)

For this project, the main SID is the BSU athletic department, which will likely need to approve the use of PracStat to record stats for volleyball practices. We don't foresee this being a huge deal as coach Joel Walton has a say in how he records statistics for the volleyball team. From what we have learned from Joel, he has not been particularly attached to any one specific stat recording application. This is a good sign, as the transition to PracStat will most likely not present a huge problem for the athletic department.

Dependencies

Optimally, voice testing will be done in a populated gymnasium, but if one is not available, background noise can be simulated. All developer laptops have microphones, so microphone access will not be an issue. The main constraint we foresee will be meeting the deadlines for test code completion.

Risks/Assumption

Delay in delivery of test items might require scheduling additional team meetings to meet the delivery date. Failure to meet individual responsibilities regarding test cases and the test plan may result in other team members being assigned additional tasks.

Tools

- JUnit will be used for unit testing.
- If necessary, an Excel spreadsheet will be used for defect tracking.

User Manual
for

PracStat

Version 1.0

Prepared by Adam Wessel, Alex Bobay, Dexter Edwards

W.E.B. Enterprises

April 22, 2019

1. Introduction

1.1 The Hardware and System Software Requirements

PracStat requires certain hardware and software in order to run properly. PracStat is designed to run on a Windows machine (either a laptop or a desktop computer). Additionally, to make use of PracStat's voice recognition functionality, a microphone is required. This microphone can be a default built-in microphone that comes with the majority of laptops and computers nowadays. Java must also be installed in order to run PracStat. Microsoft Excel is highly recommended to view generated spreadsheets.

1.2 Overview of Project Functionality

The most recent and up-to-date version of PracStat is version 1.0. This is the first generally available deployment of the PracStat software developed by W.E.B. Enterprises. Additional functionality, bug fixes, updates, etc. may be implemented in future version releases of PracStat.

1.3 Organization of User Manual

This user manual for PracStat contains all the information necessary for any user of the system (coach, statistician, or player) to get started with PracStat and utilize the application effectively and efficiently. Sections 1 and 2 of this document - Introduction and PracStat - contain general information about the current version of PracStat and it's parts. Sections 3 and 4 - Installing PracStat and Using PracStat - contain more in-depth, step-by-step information to help users use the PracStat application to the fullest. These sections go over installation, program usage, frequently encountered problems and solutions, and more.

2. PracStat System

2.1 Subsystems of the System

The PracStat application consists of 3 subsystems. These subsystems include:

- Team/Player Setup and Login
- Voice Recognition Stat Keeping
- Spreadsheet Generation

See section 2.3 - Use Cases in Each Subsystem - for more information about each specific subsystem.

2.2 Databases

To make it possible for PracStat to be used *without* any internet connection, no databases are used. Instead, a comma separated value (CSV) file is utilized. This requires no additional setup by the user. The CSV file(s) are generated and updated entirely by the PracStat application.

2.3 Use Cases in Each Subsystem

Subsystem 1: Team/Player Setup and Login

Use Cases:

- 2: Setup Team
- 3: Update Team Data
- 4: Setup Practice
- 5: Update Single Practice Info

Subsystem 2: Voice Recognition Stat Keeping

Use Cases:

- 8: Stat Entry
- 9: Error Correction

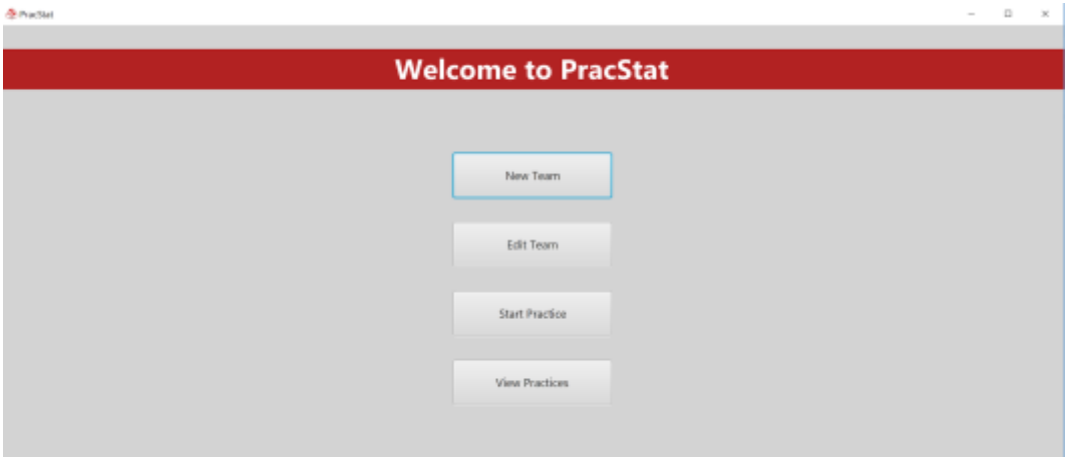
Subsystem 3: Spreadsheet Generation

Use Cases:

- 1: Generate Excel Spreadsheet
- 10: User Views Single Practice Stat
- 11: User Views Average Stats

2.4 The Interfaces Between Subsystems

The Main Menu page acts as the interface between all three subsystems. The Main Menu page is shown below:



From each of these PracStat windows (New Team, Edit Team, Start Practice, View Practices), you are able to navigate back to this Main Menu page.

3. Installing PracStat

3.1 Installation Procedure

As of now, PracStat has to be installed from a flash drive which contains the necessary files to run PracStat on your own Windows computer. Additionally, in order for a smooth installation procedure, we highly encourage that PracStat users download Java onto their machine. A detailed step-by-step installation procedure is below:

1) If Java is not already installed on your Windows computer, visit https://www.java.com/en/download/ and follow the instructions to download the latest version of Java onto your machine.
2) Using the flash drive with the necessary PracStat files, copy the files into a new folder on your computer. (Name the folder PracStat for future reference.) PracStat is now on your computer and ready to use.
3) To run PracStat, navigate to your newly made (PracStat) folder and double-click on the Executable Jar File named "PracStat". The PracStat start menu should open.
NOTE: It is highly recommended that PracStat users have Microsoft Excel installed on their computer to view the generated spreadsheets.

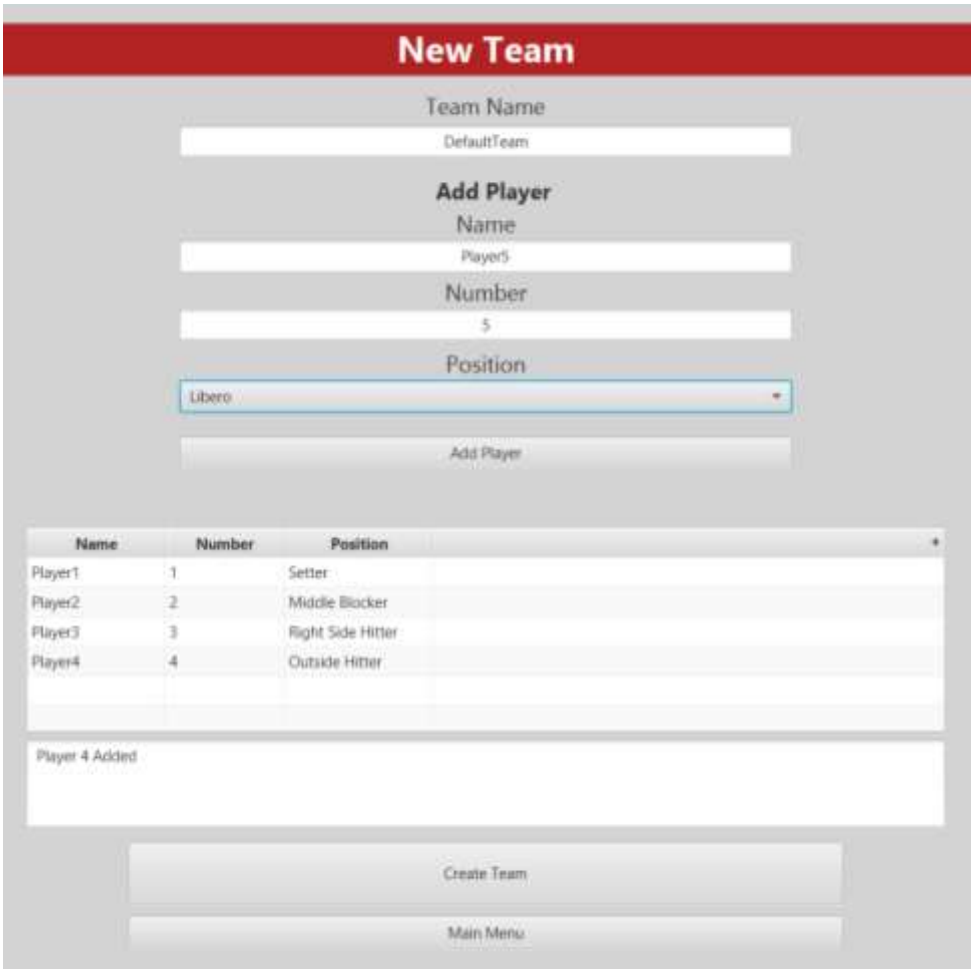
3.2 Frequently Encountered Problems and Solutions

Problem 1: "When I click the "PracStat" Executable Jar File, nothing happens."
Possible Solution(s):
- Make sure that Java is downloaded/installed on your machine. To do this, navigate to your "C:\Program Files" folder and check that it contains a subfolder named "Java".

Problem 2: "PracStat opens, but it won't let me navigate to all pages (New Team page, Edit Team page, Start Practice page, and View Practices page)."
Possible Solution(s):
- In the same folder as the "PracStat" Executable Jar File, make sure that there exists an XML document called "config". If not, you need to obtain this file from the flash drive that contains PracStat.

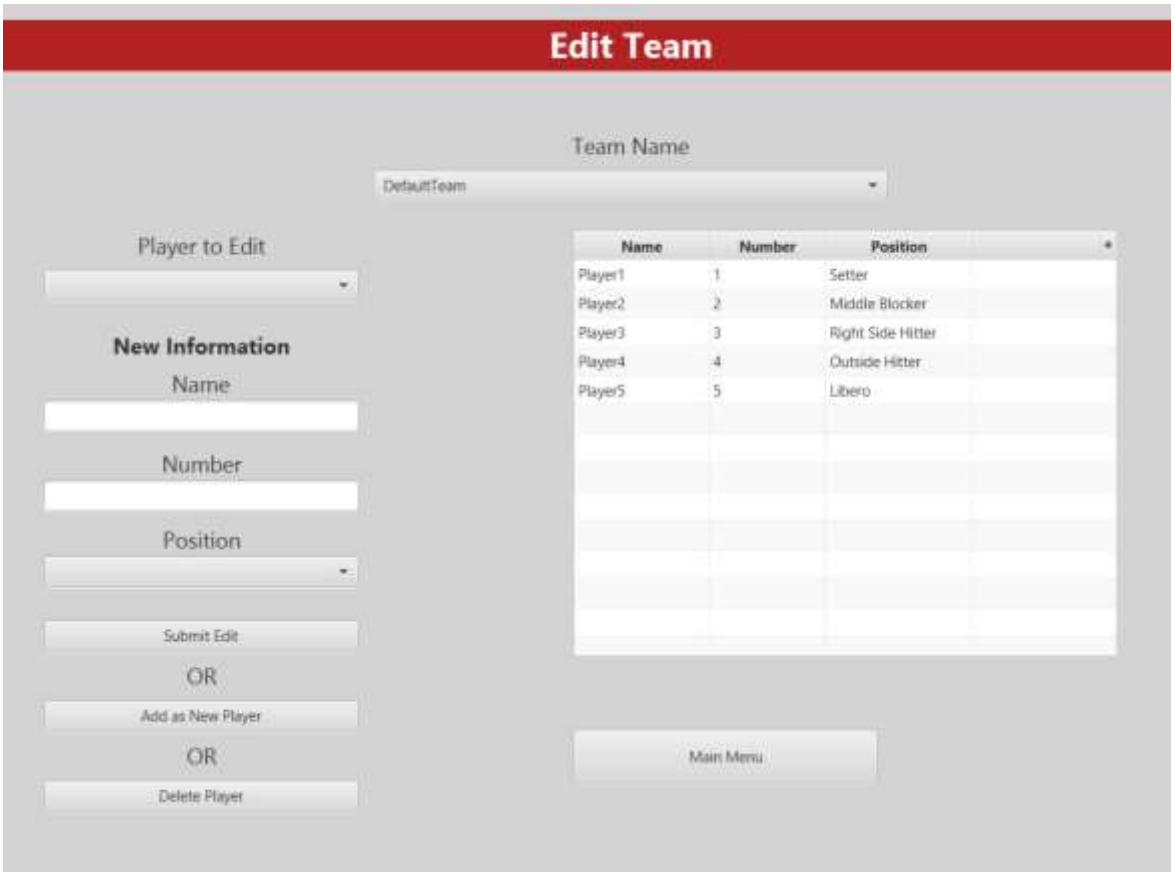
4. Using PracStat

4.1 Creating a Team



1. From the main menu, select the "New Team" option and enter the coach password.
2. Enter the name of the new team into the Team Name field. PracStat does not support duplicate team names, so this must be unique.
3. For each player on the team, enter their name, number, and position into the appropriate fields. "Name" must be at least 2 characters and "Number" must be 0-79. After all three are entered, click "Add Player" and the player will be added to the player table below.
4. When all players are included in the table, click "Create Team".

4.2 Editing a Team



1. From the main menu, select the "Edit Team" option and enter the coach password.
2. Select the team you wish to edit from the "Team Name" dropdown box.

Editing an Existing Player

1. Select the player you wish to edit from the "Player to Edit" dropdown box.
2. Enter the new name, number, and position for the player into the respective fields. Name must be at least 2 characters and number must be 0-79.
3. Click "Submit Edit"

Adding a New Player

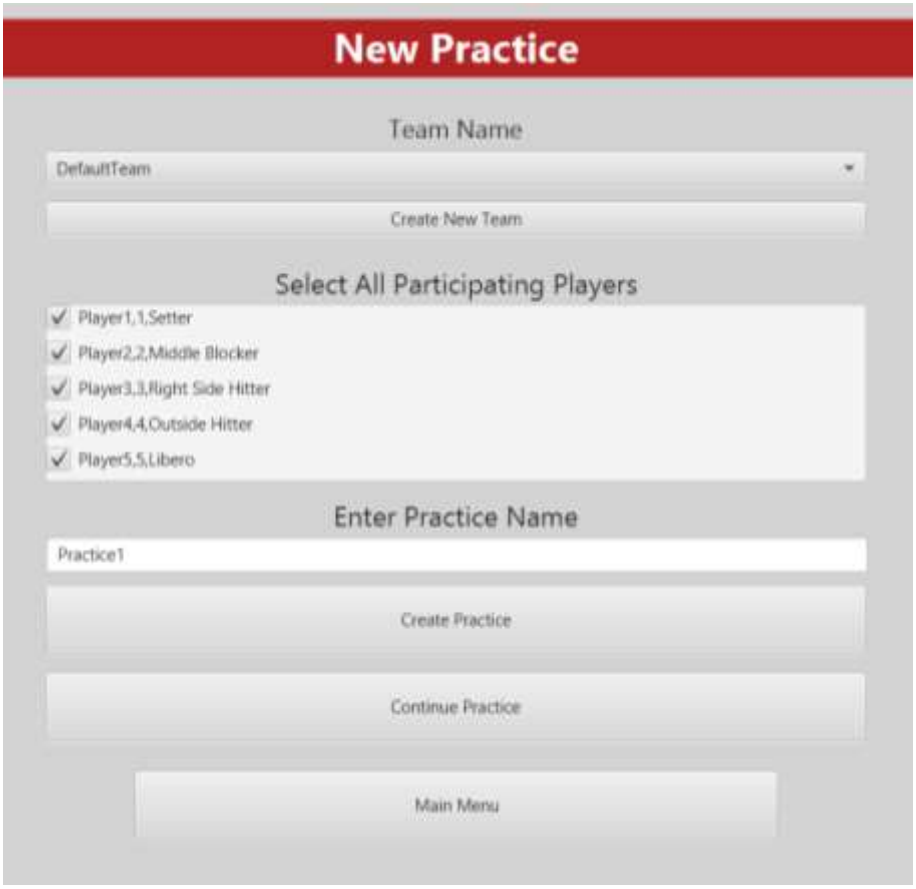
1. Enter the new player's name, number, and position into the respective fields. Name must be at least 2 character and number must be 0-79.
2. Click "Add as New Player"

Deleting an Existing Player

1. Select the player you wish to delete from the "Player to Edit" dropdown box.

2. Click “Delete Player”

4.3 Creating/Continuing a Practice



From the main menu, select the “Start Practice” option and enter the coach or statistician password.

Creating a Practice

1. Select the team you wish to create a practice for from the “Team Name” dropdown box.
2. Select the players that will be participating in this practice by checking the box next to their name in the “Select All Participating Players” box.
3. Enter the name for the new practice into the “Enter Practice Name” field.
4. Click “Create Practice”

Continuing a Practice

1. Click “Continue Practice”
2. Select the team and practice you wish to continue from the appropriate dropdown boxes.

4.4 Recording Practice Statistics

1. Once the "Create Practice" button is clicked on the New Practice page, the stat-keeping screen will appear (as displayed above).
2. The name and number of all the participating players for the practice will be displayed on the left side of the screen next to their row of statistics. By default, all player stats begin at 0.
3. To manually record statistics, simply click the up/down arrow next to the stat spinner you wish to increment/decrement.
4. To record statistics verbally, start by clicking the button that says "Click to Start Speech Recognition."
5. Once the voice recognizer loads, the button will turn green and you will hear a chime sound which indicates that you may start speaking voice commands.
6. All possible voice commands are explained in **4.6 Voice Commands**.
7. NOTE: To pause the voice recognition feature of the app, click the green voice recognition button again. This will turn the button red, and an "Ignoring Speech Recognition Results" notification will appear. To turn voice recognition back on, click the button again.
8. Once the practice is completed, and there are no more statistics to be recorded, reference the following section: **4.5 Saving Statistics**.

4.5 Saving Statistics

Once the practice is completed and all stats have been recorded, navigate to the Edit menu bar in the top left corner of the page. Click Edit -> Save Stats. If successful, you will receive a pop-up alert notifying you. Additionally, once you click Edit -> Save Stats, the statistics for that particular practice will also be added to the Season Stats which can be found under Practices as well. The Season Stats "practice" allows coaches, statisticians, and players to view statistics from the entire season (all practices).

NOTE: Practice statistics are saved to Season Stats only if the number of players in that particular practice equals the number of players on the entire team. Additionally, for the Season Stats to be accurate, PracStat assumed that you *only* click Edit -> Save Stats *once* per practice once all stats are accurately recorded.

4.6 Voice Commands

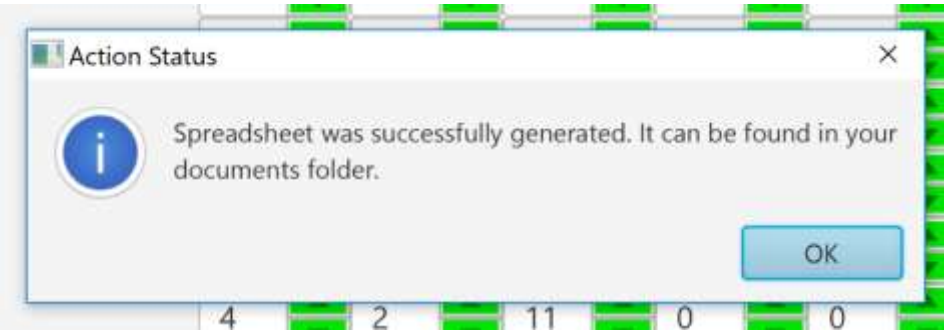
All voice commands are of the form <Number> <Statistic>. First, you say the number of the player you'd wish to give a stat too (e.g., "one", "seven", "nineteen", etc.). Then immediately following that, you say a statistic (e.g., "dig", "block", "service error", etc.). A complete list of possible statistics are shown below. Note that by default all voice commands increment a statistic by 1.

-**kill** (to record a kill)
-**attack error** (to record an attack error)
-**attack** (to record an attack attempt)
-**assist** (to record an assist)
-**ace** (to record a service ace)
-**service ace** (to record a service ace)
-**serve ace** (to record a service ace)
-**ace serve** (to record a service ace)
-**service error** (to record a service error)
-**serve error** (to record a service error)
-**receive error** (to record a receive error)
-**dig** (to record a dig)
-**solo block** (to record a solo block)
-**block** (to record a solo block)
-**block assist** (to record a block assist)
-**block error** (to record a block error)
-**lift** (to record a ball handling error)
-**double** (to record a ball handling error)
-**double contact** (to record a ball handling error)

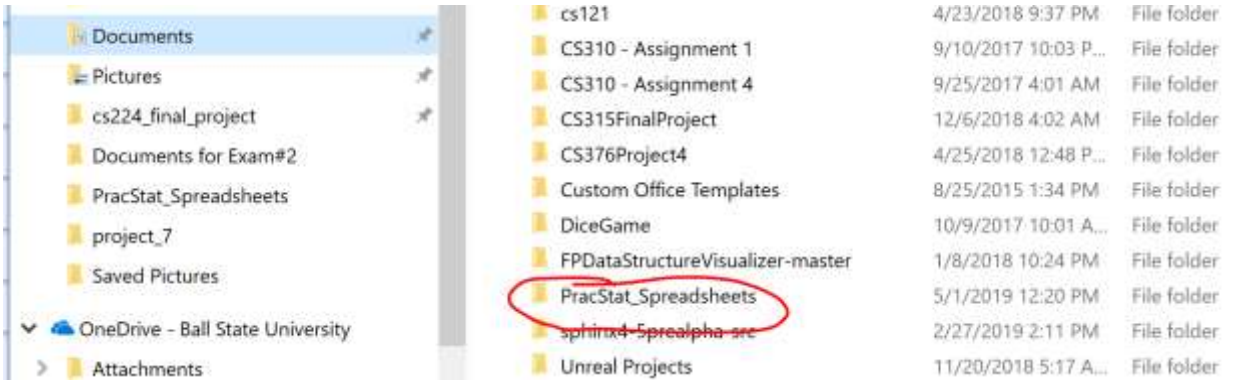
Each spoken voice command (the player number and the stat command) will increment the corresponding statistic automatically. If the voice recognizer successfully recognizes the voice command, the correct statistic in the table (shown above) will turn red for one second and increment by 1, visually showing you that the command has worked. If the voice command fails to be recognized by the voice recognizer, then no statistic will be incremented and the words that the voice recognizer recognized will appear near the top right corner of the screen.

4.7 Generating Spreadsheets

PracStat enables coaches, statisticians, and players to generate spreadsheets from any and all practice sessions. Once the statistics from a particular practice are saved (see section **4.5 Saving Statistics**), you may generate a spreadsheet that displays all of the statistical information as well as hitting percentages, total points, etc. To do this, on the main practice page, navigate to the File menu bar in the top left corner of the screen and click on File -> Generate Spreadsheet. If successful, you will receive a pop-up alert that notifies you that the newly created spreadsheet is in your documents folder.



Using your Windows explorer, navigate to Documents folder and find the newly generated folder named “PracStat_Spreadsheets” as shown below:



This folder will hold all of your generated Microsoft excel spreadsheets. They are automatically named using the format: <Team Name><Practice Name><Date Generated> so that you can find your spreadsheets easily. To open a spreadsheet in excel, simply double click the file. A final spreadsheet is pictured below:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	BSU 19	Practice 1														
2	#	PLAYER	K	E	TA	PCT	AST	SA	SE	RE	DIG	BS	BA	BE	BH	PTS
3	1	Wessel	0	0	0	0	2	0	0	1	5	0	0	0	0	0
4	2	Jenness	0	0	0	0	0	0	1	0	2	0	0	0	0	0
5	3	Chinnici	0	0	0	0	0	0	2	0	1	0	0	0	0	0
6	4	Lavanchy	0	0	0	0	0	0	0	0	4	0	0	0	1	0
7	5	Isaacson	0	0	0	0	0	2	0	0	2	0	0	0	0	2
8	6	Ensalaco	0	0	0	0	2	1	0	1	1	0	0	0	0	1
9	7	Nielsen	4	2	11	0.182	0	0	0	0	1	1	1	0	0	5.5
10	8	Reardon	0	0	0	0	0	1	0	0	1	0	0	0	0	1
11	9	Swartz	1	0	4	0.25	0	0	0	0	0	0	5	0	0	3.5
12	10	Amos	0	0	2	0	0	0	2	0	1	0	4	0	0	2
13	11	Egharevba	1	1	3	0	0	1	0	0	0	1	0	1	0	3
14	12	Romano	0	1	1	-1	15	0	0	0	1	0	0	0	1	0
15	14	Szews	9	2	16	0.438	0	0	1	0	0	0	0	0	0	9
16	15	Hippe	2	0	6	0.333	0	0	0	0	0	0	0	0	0	2
17	17	Dorgan	0	2	3	-0.667	0	0	1	0	0	0	0	0	0	0
18	18	Shepherd	0	1	4	-0.25	0	0	0	0	3	0	0	0	0	0
19	19	Turner	3	1	6	0.333	0	0	0	0	1	0	0	0	0	3
20	20	Siebum	5	2	13	0.231	0	1	2	0	0	0	0	0	0	6
21	21	Scharenborg	0	0	0	0	4	0	1	0	0	0	0	0	0	0
22	22	Martinski	1	0	2	0.5	0	0	0	0	2	0	0	0	0	1
23		Totals	26	12	71	0.197	23	6	10	2	25	2	10	1	2	39

NOTE: Also, on the File menu bar in the top left corner of the screen, there is a File -> Generate Average Spreadsheet button. This will have the same procedure as above; however, the statistics generated in this spreadsheet will be the *averages* taken from each practice.

4.8 Closing PracStat

Once completed with PracStat for the time being, simply click the “X” in the top right hand corner of the screen. You will receive the pop-up alert shown below:

If you saved all of the practice stats already. (See section **4.5 Saving Statistics)**. Then you are good to click "OK" to exit PracStat.

NOTE: Exit PracStat and re-open the application between each practice. This will ensure that the voice recognition capability continues to work correctly.

References

Opia, Chuks E. “How to Build a Simple Speech Recognition App.” Free Code Camp, 27 Mar. 2018, medium.freecodecamp.org/how-to-build-a-simple-speech-recognition-app-a65860da6108.

Pressman, Roger S. Software Engineering: a Practitioner's Approach. McGraw-Hill Education, 2015.

“Where Developers Learn, Share, & Build Careers.” *Stack Overflow,* stackoverflow.com.

Appendix A

Ball State University, Computer Science Capstone Course, CS 495 and CS 498

CLIENT ACCEPTANCE FORM

Purpose: This form is completed at the end of the team engagement where the team is handing over deliverables to the client partner(s) associated with the team project. The form verifies what deliverables are being turned over to the client partner(s) and that the client partner(s) has accepted / approved those deliverables.

Project Name	PracStat		
Project Sponsor	Joel Walton		
Team : <name>	W.E.B Enterprises		
	Adam Wessel	atwessel@bsu.edu	262-894-3642
	Dexter Edwards	djedwards2@bsu.edu	317-331-1044
	Alex Babay	ambobay2@bsu.edu	260-437-3056
	(name)	(email)	(phone number)
Project Description	Voice activated statistic tracking software developed for the Ball State men's volleyball team.		

Important Notes for Completing this Document

Each section of the Client Acceptance Form must be completed in full. If a particular section is not applicable to this project, then you must write *Not Applicable* and provide a reason.

Important Note: No sections are to be deleted from this document. Text contained within << >> provides information on how to complete that section and can be deleted once the section has been completed.

LIST OF DELIVERABLES COMPLETED

Deliverables	PracStat Version 1.0 PracStat installation PracStat User manual
Acceptance Response	☒ Accepted ☐ Not Accepted until below issues are addressed ☐ Accepted provided below issues are addressed
Issues	N/A
Additional Comments	N/A

PREPARED BY

Project Manager			
	(name)	(signature)	(date)

APPROVALS

Sponsor	Joel B Walton	Joel B. Walton	5-2-2019
	(name)	(signature)	(date)
Application Delivery Manager			
	(name)	(signature)	(date)